Technical Report

NetApp Data Compression and Deduplication Deployment and Implementation Guide
Clustered Data ONTAP

Sandra Moulton, NetApp
February 2014 | TR-3966

Abstract

This technical report focuses on clustered Data ONTAP® implementations of NetApp® deduplication and NetApp data compression. For information on implementation with Data ONTAP 8.1 operating in 7-Mode, refer to TR-3958: NetApp Data Compression and Deduplication Deployment and Implementation Guide, Data ONTAP 8.1 Operating in 7-Mode.

This report describes in detail how to implement and use both technologies and provides information on best practices, operational considerations, and troubleshooting.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1   Introduction

One of the biggest challenges for companies today continues to be the cost of storage. Storage represents the largest and fastest growing IT expense. NetApp's portfolio of storage efficiency technologies is aimed at lowering this cost. NetApp deduplication and data compression are two key components of NetApp storage efficiency technologies that enable users to store the maximum amount of data for the lowest possible cost.

This paper focuses on two NetApp features: NetApp deduplication as well as NetApp data compression. These technologies can work together or independently to achieve optimal savings. NetApp deduplication is a process that can be scheduled to run when it is most convenient, while NetApp data compression has the ability to run either as an inline process as data is written to disk or as a scheduled process. When the two are enabled on the same volume, the data is first compressed and then deduplicated. Deduplication will remove duplicate compressed or uncompressed blocks in a data volume. Although compression and deduplication work well together, it should be noted that the savings will not necessarily be the sum of the savings when each is run individually on a dataset.

Notes:

1. Whenever references are made to deduplication in this document, you should assume we are referring to NetApp deduplication.
2. Whenever references are made to compression in this document, you should assume we are referring to NetApp data compression.
3. Unless otherwise mentioned, when references are made to compression they are referring to postprocess compression. References to inline compression are referred to as "inline compression."
4. The same information applies to both FAS and V-Series systems, unless otherwise noted.
5. An Infinite Volume is a single large scalable file system that contains a collection of FlexVol® volumes called constituents.

   Whenever a reference is made to an Infinite Volume this refers to the logical container, not the individual constituents.
6. An Infinite Volume includes a namespace constituent and multiple data constituents. The namespace constituent contains the directory hierarchy and file names with pointer redirectors to the physical location of the data files. The data constituents contain the physical data within an Infinite Volume.

   Whenever a reference is made to either the namespace constituent or data constituents, this refers to that specifically, not the entire Infinite Volume itself.
7. Whenever references are made to FlexVol volumes these are specific to Data ONTAP 8.1.
8. Whenever references are made to volumes these are applicable to both FlexVol volumes as well as Infinite Volumes.
9. As the title implies, this technical report covers clustered Data ONTAP 8.1 and above. There is an equivalent technical report for Data ONTAP 8.1 and 8.2 operating in 7-Mode: TR-3958: NetApp Data Compression and Deduplication Deployment and Implementation Guide for Data ONTAP 8.1 and 8.2 Operating in 7-Mode.

For more information on Infinite Volumes, see TR-4178: NetApp Infinite Volume Deployment and Implementation Guide.

2   NetApp Deduplication

Part of NetApp's storage efficiency offerings, NetApp deduplication provides block-level deduplication within a FlexVol volume or data constituent. NetApp V-Series is designed to be used as a gateway system that sits in front of third-party storage, allowing NetApp storage efficiency and other features to be used on third-party storage.

Essentially, deduplication removes duplicate blocks, storing only unique blocks in the FlexVol volume or data constituent, and it creates a small amount of additional metadata in the process. Notable features of deduplication include:

- It works with a high degree of granularity: that is, at the 4KB block level.
- It operates on the active file system of the FlexVol volume or data constituent. Any block referenced by a Snapshot™ copy is not made "available" until the Snapshot copy is deleted.
- It is a background process that can be configured to run automatically, be scheduled, or run manually through the command line interface (CLI). NetApp Systems Manager, or NetApp OnCommand® Unified Manager.
- It is application transparent, and therefore it can be used for deduplication of data originating from any application that uses the NetApp system.
- It is enabled and managed by using a simple CLI or GUI such as Systems Manager or NetApp OnCommand Unified Manager.

Figure 1) How NetApp deduplication works at the highest level.
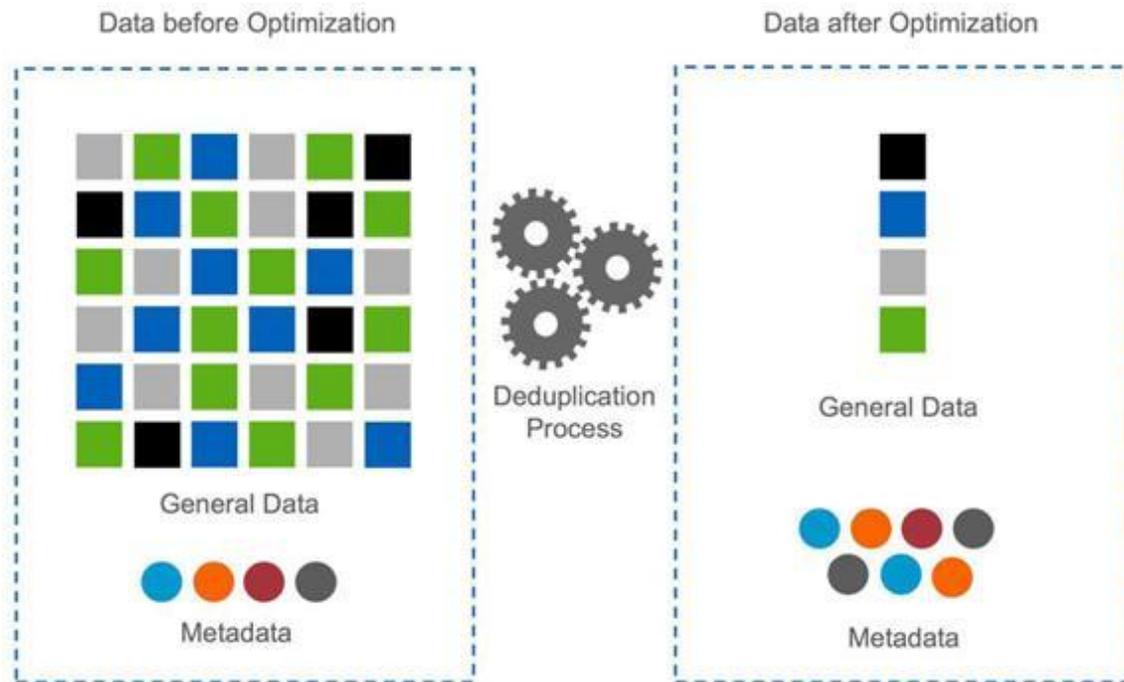
In summary, this is how deduplication works: Newly saved data is stored in 4KB blocks as usual by Data ONTAP. Each block of data has a digital fingerprint, which is compared to all other fingerprints in the FlexVol volume or data constituent. If two fingerprints are found to be the same, a byte-for-byte comparison is done of all bytes in the block. If there is an exact match between the new block and the existing block on the FlexVol volume or data constituent, the duplicate block is discarded and its disk space is reclaimed.

2.1 Deduplicated FlexVol Volumes and Infinite Volume Data Constituents

A deduplicated volume is a FlexVol volume or data constituent that contains shared data blocks. Data ONTAP supports shared blocks in order to optimize storage space consumption. Basically, in one FlexVol volume or data constituent, there is the ability to have several references to the same data block.

In Figure 2, the number of physical blocks used on the disk is 3 (instead of 6), and the number of blocks saved by deduplication is 3 (6 minus 3). In this document, these are referred to as used blocks and saved blocks.

Figure 2) Data structure in a deduplicated FlexVol volume or data constituent.
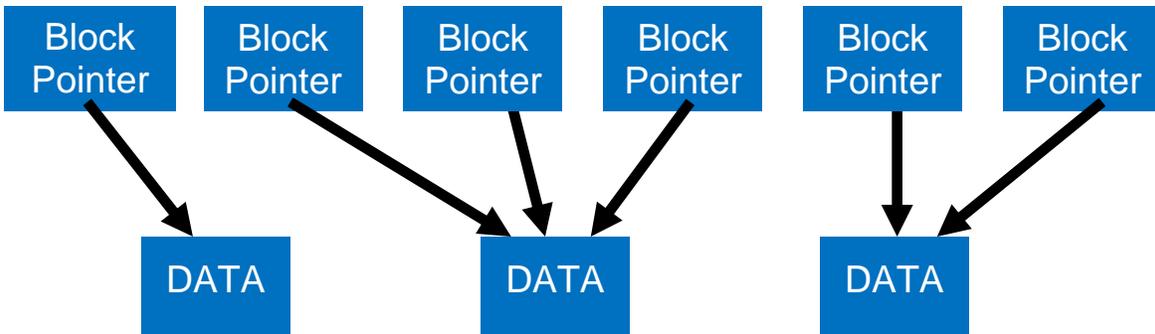
Each data block has a reference count that is kept in the volume or data constituent metadata. In the process of sharing the existing data and eliminating the duplicate data blocks, block pointers are altered. For the block that remains on disk with the block pointer, its reference count will be increased. For the block that contained the duplicate data, its reference count will be decremented. When no block pointers have reference to a data block, the block is released.

NetApp deduplication technology allows duplicate 4KB blocks anywhere in the FlexVol volume or data constituent to be deleted, as described in the following sections.

The maximum sharing for a block is 32,767. This means, for example, that if there are 64,000 duplicate blocks, deduplication would reduce that to only 2 blocks.

2.2 Deduplication Metadata

The core enabling technology of deduplication is fingerprints. These are unique digital "signatures" for every 4KB data block in the FlexVol volume or data constituent.

When deduplication runs for the first time on a volume with existing data, it scans the blocks in the FlexVol volume or data constituent and creates a fingerprint database that contains a sorted list of all fingerprints for used blocks in the FlexVol volume or data constituent.

After the fingerprint file is created, fingerprints are checked for duplicates, and, when duplicates are found, a byte-by-byte comparison of the blocks is done to make sure that the blocks are indeed identical. If they are found to be identical, the block's pointer is updated to the already existing data block, and the new (duplicate) data block is released.

Releasing a duplicate data block entails updating the block pointer, incrementing the block reference count for the already existing data block, and freeing the duplicate data block. In real time, as additional data is written to the deduplicated volume or data constituent, a fingerprint is created for each new block and written to a change log file. When deduplication is run subsequently, the change log is sorted, its sorted fingerprints are merged with those in the fingerprint file, and then the deduplication processing occurs.

There are two change log files, so that as deduplication is running and merging the fingerprints of new data blocks from one change log file into the fingerprint file, the second change log file is used to log the fingerprints of new data written to the FlexVol volume or data constituent during the deduplication process. The roles of the two files are then reversed the next time that deduplication is run. (For those familiar with Data ONTAP usage of NVRAM, this is analogous to when it switches from one half to the other to create a consistency point.)

Here are some additional details about the deduplication metadata.

- There is a fingerprint record for every 4KB data block, and the fingerprints for all the data blocks in the FlexVol volume or data constituent are stored in the fingerprint database file. Starting in Data ONTAP 8.2, this will only be for each 4K data block physically present in the volume as opposed to each 4KB logically in the volume.

- Fingerprints are not deleted from the fingerprint file automatically when data blocks are freed. When a threshold of the number of new fingerprints is 20% greater than the number of data blocks used in the volume, the stale fingerprints are deleted. This can also be done by a manual operation using the advanced mode command `volume efficiency check`.

- Starting with Data ONTAP 8.1.1, the change log file size limit is set to 1% of the volume size except if it is a SnapVault® destination, and then the change log file size limit is set to 4% of the volume size. The change log file size is relative to the volume size limit. The space assigned for the change log in a volume is not reserved.

- The deduplication metadata for a FlexVol volume or data constituent is located inside the aggregate, and a copy of this will also be stored in the FlexVol volume or data constituent. The copy inside the aggregate is used as the working copy for all deduplication operations. Change log entries will be appended to the copy in the FlexVol volume or data constituent.

- During an upgrade of a major Data ONTAP release such as 8.1 to 8.2, the fingerprint and change log files are automatically upgraded to the new fingerprint and change log structure the first time volume efficiency operations start after the upgrade completes. Be aware that this is a one-time operation, and it can take a significant amount of time to complete, during which time you can see an increased amount of CPU on your system. You can see the progress of the upgrade using the `volume efficiency show –instance` command.

- The deduplication metadata requires a minimum amount of free space in the aggregate equal to 3% of the total amount of data for all deduplicated FlexVol volumes or data constituents within the aggregate. Each FlexVol volume or data constituent should have 4% of the total amount of data's worth of free space, for a total of 7%. For Data ONTAP 8.1 the total amount of data should be calculated using the total amount of logical data. Starting with Data ONTAP 8.2 the total amount of data should be calculated based on the total amount of physical data.

- The deduplication fingerprint files are located inside both the volume or data constituent and the aggregate. This allows the deduplication metadata to follow the volume or data constituent during operations such as a volume SnapMirror® operation. If the volume or data constituent ownership is changed because of a disaster recovery operation with volume SnapMirror, the next time deduplication is run it will automatically recreate the aggregate copy of the fingerprint database from the volume or data constituent copy of the metadata. This is a much faster operation than recreating all the fingerprints from scratch.

Deduplication Metadata Overhead

Although deduplication can provide substantial storage savings in many environments, a small amount of storage overhead is associated with it. The deduplication metadata for a FlexVol volume or data constituent is located inside the aggregate, and a copy of this will also be stored in the FlexVol volume or data constituent.

The guideline for the amount of extra space that should be left in the FlexVol volume or data constituent and aggregate for the deduplication metadata overhead is as follows.

- Volume or data constituent deduplication overhead.

    For Data ONTAP 8.1, each FlexVol volume or data constituent with deduplication enabled, up to 4% of the logical amount of data written to that volume or data constituent is required in order to store volume or data constituent deduplication metadata.

    Starting with Data ONTAP 8.2 each FlexVol volume or data constituent with deduplication enabled, up to 4% of the physical amount of data written to that volume or data constituent is required in order to store FlexVol volume or data constituent deduplication metadata. This value will never exceed the maximum FlexVol volume or data constituent size times 4%.

- Aggregate deduplication overhead.

    For Data ONTAP 8.1, each aggregate that contains any volumes or a data constituents with deduplication enabled, up to 3% of the logical amount of data contained in all of those volumes or data constituent with deduplication enabled within the aggregate is required in order to store the aggregate deduplication metadata.

    Starting with Data ONTAP 8.2, each aggregate that contains any FlexVol volumes or data constituents with deduplication enabled, up to 3% of the physical amount of data contained in all of those FlexVol volumes or data constituents with deduplication enabled within the aggregate is required in order to store the aggregate deduplication metadata.

Deduplication Metadata Overhead Examples: Data ONTAP 8.1
For example, if 100GB of data is to be deduplicated in a single FlexVol volume, then there should be 4GB of available space in the volume and 3GB of space available in the aggregate. As a second example, consider a 2TB aggregate with four volumes, each 400GB in size, in the aggregate. Three volumes are to be deduplicated, with 100GB of data, 200GB of data, and 300GB of data, respectively. The volumes need 4GB, 8GB, and 12GB of space, respectively, and the aggregate needs a total of 18GB ((3% of 100GB) + (3% of 200GB) + (3% of 300GB) = 3+6+9=18GB) of space available in the aggregate.

Deduplication Metadata Overhead Examples: Data ONTAP 8.2
For example, if you have 100GB of logical data and get 50GB of savings in a single FlexVol volume, then you will have 50GB of physical data. Given this, there should be 2GB (4% of 50GB) of available space in the volume and 1.5GB of space available in the aggregate. As a second example, consider a 2TB aggregate with four volumes, each 400GB in size, in the aggregate. Three volumes are to be deduplicated, with the following:
    Vol1: 100GB of logical data, which will have 50% savings (50% of 100GB) = 50GB physical data
    Vol2: 200GB of logical data, which will have 25% savings (75% of 200GB) = 150GB physical data
    Vol3: 300GB of logical data, which will have 75% savings (25% of 300GB) = 75GB physical data
The required amount of space for deduplication metadata is as follows:
    Vol1: 2GB (50GB * 4%)
    Vol2: 6GB (150GB * 4%)
    Vol3: 3GB (75GB * 4%)
The aggregate needs a total of 8.25GB ((3% of 50GB) + (3% of 150GB) + (3% of 75GB)) = 1.5+4.5+2.25=8.25GB) of space available in the aggregate.

The primary fingerprint database, otherwise known as the working copy, is located outside the volume or data constituent, in the aggregate, and is therefore not captured in Snapshot copies. The change log files and a backup copy of the fingerprint database are located within the volume or data constituent and are therefore captured in Snapshot copies. Having the primary (working) copy of the fingerprint database outside the volume or data constituent enables deduplication to achieve higher space savings. However, the other temporary metadata files created during the deduplication operation are still placed inside the volume or data constituent. These temporary metadata files are deleted when the deduplication operation is complete. However, if Snapshot copies are created during a deduplication operation, these temporary metadata files can get locked in Snapshot copies, and they remain there until the Snapshot copies are deleted.

3 NetApp Data Compression

NetApp data compression is a software-based solution that provides transparent data compression. It can be run inline or postprocess and also includes the ability to perform compression of existing data. No application changes are required to use NetApp data compression.

3.1 How NetApp Data Compression Works

NetApp data compression does not compress the entire file as a single contiguous stream of bytes. This would be prohibitively expensive when it comes to servicing small reads or overwrites from part of a file because it requires the entire file to be read from disk and uncompressed before the request can be served. This would be especially difficult on large files. To avoid this, NetApp data compression works by compressing a small group of consecutive blocks, known as a compression group. In this way, when a read or overwrite request comes in, we only need to read a small group of blocks, not the entire file. This optimizes read and overwrite performance and allows greater scalability in the size of the files being compressed.

Compression Groups

The NetApp compression algorithm divides a file into "compression groups." The file must be larger than 8k or it will be skipped for compression and written to disk uncompressed. Compression groups are a maximum of 32K. A compression group contains data from one file only. A single file can be contained within multiple compression groups. If a file were 60k it would be contained within two compression groups. The first would be 32k and the second 28k.

Compressed Writes

NetApp handles compression write requests at the compression group level. Each compression group is compressed separately. The compression group is left uncompressed unless a savings of at least 25% can be achieved on a per-compression-group basis; this optimizes the savings while minimizing the resource overhead.

Figure 3) Compression write request handling.
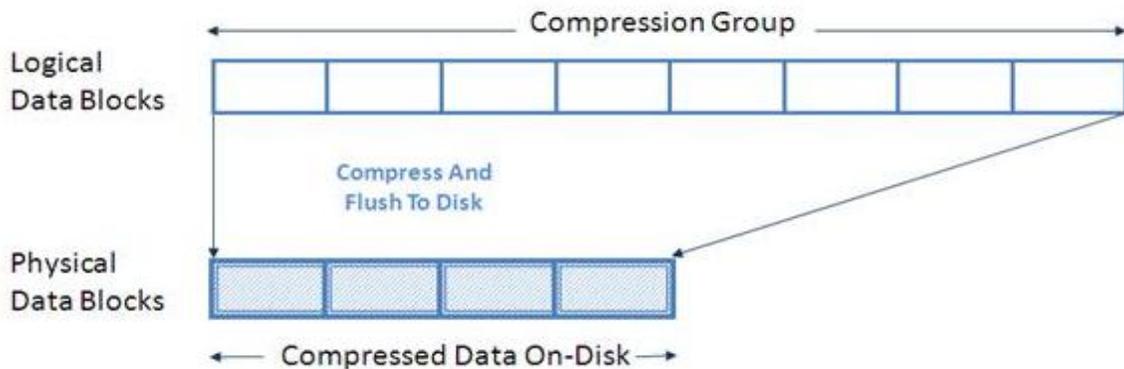
Since compressed blocks contain fewer blocks to be written to disk, compression will reduce the amount of write I/Os required for each compressed write operation. This will not only lower the data footprint on disk but can also decrease the time to complete your backups; see the "Feature Interoperability" section for details on volume SnapMirror.

Compressed Reads

When a read request comes in, we read only the compression group(s) that contain the requested data, not the entire file. This optimizes the amount of I/O being used to service the request. When reading compressed data, only the required compression group data blocks will be transparently decompressed in memory. The data blocks on disk will remain compressed. This has much less overhead on the system resources and read service times.

In summary, the NetApp compression algorithm is optimized to reduce overhead for both reads and writes.

3.2 When Data Compression Runs

Inline Operations

NetApp data compression can be configured as an inline operation. In this way, as data is sent to the storage system it is compressed in memory before being written to the disk. The advantage of this implementation is that it can reduce the amount of write I/O. This implementation option can affect your write performance and thus should not be used for performance-sensitive environments without first doing proper testing to understand the impact.

In order to provide the fastest throughput, inline compression will compress most new writes but will defer some more performance-intensive compression operations to compress when the next postprocess compression process is run. An example of a performance-intensive compression operation is a small (<32k) partial file overwrite.

Starting in Data ONTAP 8.2, we added the idd (incompressible data detection) option to inline compression. If this option is set to true (default is false), then the way it works depends on whether you are writing a large or small file. By default we assume a file that is less than 500MB is a small file. You can modify the size of what is considered a small or large file by modifying the `quick-check-fsize` option of the `volume efficiency config` command.

For small files we compress each compression group within the file unless or until we find one that has less than 25% savings. At that time we mark the file as incompressible and stop any additional inline compress attempts. For large files we try to compress the first 4K of each compression group. If there is at least 25% savings in the first 4K, we continue compressing the rest of the compression group. If the first 4K of a compression group does not have at least 25% savings, we do not continue inline compression attempts on the rest of the compression group and write it as uncompressed. We continue to do this quick check on all compression groups within the file.

Postprocess compression will continue to try to compress any compression groups that were skipped by inline compression.

This can be especially beneficial for customers who have a large mix of highly and minimally compressible data in their volumes that they want to compress with inline compression.

Postprocess Operations

NetApp data compression includes the ability to run postprocess compression. Postprocess compression uses the same schedule as deduplication utilizes. If compression is enabled when the volume efficiency schedule initiates a postprocess operation it runs compression first, followed by deduplication. It includes the ability to compress data that existed on disk prior to enabling compression.

If both inline and postprocess compression are enabled, then postprocess compression will try to compress only blocks that are not already compressed. This includes blocks that were bypassed by inline compression such as small partial file overwrites.

4   General Compression and Deduplication Features

Both compression and deduplication are enabled on a per FlexVol volume or per-Infinite-Volume basis. They can be enabled on any number of FlexVol volumes or data constituents in a storage system. When enabling on an Infinite Volume it automatically enables it on all data constituents; you can't enable it on some data constituents and not others. While deduplication can be enabled on FlexVol volumes contained within either 32-bit or 64-bit aggregates, compression can only be enabled on FlexVol volumes contained within a 64-bit aggregate.

Compression requires that deduplication first be enabled on a volume; it can't be enabled without deduplication. Inline compression requires that both deduplication and postprocess compression also be enabled.

Compression and deduplication share the same scheduler and can be scheduled to run in one of four different ways:

- Inline (compression only)
- Scheduled on specific days and at specific times
- Manually, by using the command line
- Automatically, when 20% new data has been written to the volume
- SnapVault software based, when used on a SnapVault destination

Only one postprocess compression or deduplication process can run on a FlexVol volume or data constituent at a time. Up to eight compression/deduplication processes can run concurrently on eight different volumes or data constituents within the same NetApp storage system. If there is an attempt to run additional postprocess compression or deduplication processes beyond the maximum, the additional operations will be placed in a pending queue and automatically started when there are free processes.

Postprocess compression and deduplication processes periodically create checkpoints so that in the event of an interruption to the scan it can continue from the last checkpoint.

5   Compression and Deduplication System Requirements

This section discusses what is required to install deduplication and/or compression and details about the maximum amount of data that will be compressed and deduplicated. Although the section discusses some basic things, we assume that the NetApp storage system is already installed and running and that the reader is familiar with basic NetApp administration.

5.1 Overview of Requirements

Table 1) Overview of compression and deduplication requirements for clustered Data ONTAP.

Requirement	Deduplication	Compression
Data ONTAP version	Minimum clustered Data ONTAP 8.1	
License	Neither deduplication nor compression requires any licenses; you can simply enable on a volume-by-volume basis	
Hardware	All FAS and V-Series systems that are supported with clustered Data ONTAP	
Volume type supported	FlexVol or Infinite Volumes only, no traditional volumes	
Aggregate type supported	32-bit and 64-bit	64-bit only
Maximum volume size	Neither compression nor deduplication impose a limit on the maximum volume or data constituent size supported; therefore, the maximum volume or data constituent limit is determined by the type of storage system regardless of whether deduplication or compression is enabled	
Supported protocols	All	

Note: NetApp deduplication and data compression on V-Series systems with array LUNs are supported using block checksum scheme (BCS) and, starting in Data ONTAP 8.1.1, also using zone checksum scheme (ZCS). For more information, refer to TR-3461: V-Series Best Practice Guide.

5.2 Maximum Logical Data Size Processing Limits

In Data ONTAP 8.1, the maximum logical data size that will be processed by postprocess compression and deduplication is equal to the maximum volume size supported on the storage system regardless of the size of the volume or data constituent created. Starting in Data ONTAP 8.2, the maximum logical data size that will be processed by postprocess compression and deduplication is 640TB regardless of the size of the volume or data constituent created. Once this logical limit is reached, writes to the volume or data constituent will continue to work successfully; however, postprocess compression and deduplication will fail with the error message "maximum logical data limit has reached."

As an example in Data ONTAP 8.1, if you had a FAS6240 that has a 100TB maximum volume size and you created a 100TB volume or data constituent, the first 100TB of logical data will compress and deduplicate as normal. However, any additional new data written to the volume or data constituent after the first 100TB will not be postprocess compressed or deduplicated until the logical data becomes less than 100TB. If inline compression is enabled on the volume or data constituent it will continue to be inline compressed until the volume or data constituent is completely full. A second example in Data ONTAP 8.1 is a system with a 50TB volume size limit and you create a 25TB volume. In this case the first 50TB of logical data will compress and deduplicate as normal; however, any additional new data written to the volume after the first 50TB will not be postprocess compressed or deduplicated until the amount of logical data is less than 50TB. If inline compression is enabled on the volume it will continue to be inline compressed until the volume is completely full.

As an example in Data ONTAP 8.2, if you had a FAS6270 that has a 100TB maximum volume size and you created a 100TB volume or data constituent, the first 640TB of logical data will compress and deduplicate as normal. However, any additional new data written to the volume or data constituent after the first 640TB will not be postprocess compressed or deduplicated until the logical data becomes less than 640TB. If inline compression is enabled on the volume or data constituent, it will continue to be inline compressed until the volume or data constituent is completely full.

6 Whether to Enable Compression and/or Deduplication

Choosing when to enable compression or deduplication involves balancing the benefits of space savings against the potential overhead. Your savings and acceptable overhead will vary depending on the use case, and, as such, you may find some solutions suited for primary tier and others better suited for backup/archive tier only. The amount of system resources they consume and the possible savings highly depend on the type of data. Performance impact will vary in each environment, and NetApp highly recommends that the performance impact be fully tested and understood before implementing in production.

6.1 When Should I Consider Enabling Compression and/or Deduplication?

Table 2 shows some examples of where customers have commonly enabled compression or deduplication. These are strictly examples, not rules; your environment may have different performance requirements for specific use cases. NetApp highly recommends that the performance impact be fully tested and understood before you decide to implement in production.

Table 2) Commonly used compression and deduplication use cases.

Type of Application	Storage Efficiency Option(s) Commonly Used
Backup/Archive	Inline Compression + Postprocess Compression and Deduplication
Test/Development	Inline Compression + Postprocess Compression and Deduplication
File Services, Engineering Data, IT Infrastructure	Inline Compression + Postprocess Compression and Deduplication
Geoseismic	Inline Compression Only (set postprocess schedule to not run)
Virtual Servers and Desktops (Boot Volumes)	Deduplication Only on Primary, Inline Compression + Postprocess Compression and Deduplication on Backup/Archive
Oracle® OLTP	None on Primary, Inline Compression + Postprocess Compression and Deduplication on Backup/Archive
Oracle Data Warehouse	Deduplication[1] on Primary, Inline Compression + Postprocess Compression and Deduplication on Backup/Archive
Exchange 2010	Deduplication and Postprocess Compression[2] on Primary, Inline Compression + Postprocess Compression and Deduplication on Backup/Archive

Note: These are guidelines, not rules, and assume that the savings are high enough, your system has sufficient system resources, and any potential effect on performance is fully understood and acceptable.

---

[1] Deduplication on Oracle Data Warehouse on primary is typically only utilized where there is sufficient savings and Oracle is configured with a 16k or 32k block size. Testing should be done before implementing in production. NetApp recommends using a Flash Cache card.
[2] Compression on Exchange is a less common use case that can be utilized but only where there is sufficient time to run postprocess compression/deduplication processes, there are sufficient savings, and the performance impact is fully understood. Testing should be done before implementing in production.

6.2   When Should I Not Consider Enabling Compression and/or Deduplication?

Some examples of when not to use deduplication on a volume include:

- Savings less than the amount of deduplication metadata
- Data is being overwritten at a rapid rate

Some examples of when not to use compression on a volume include:

- Large number of small overwrites of a file (typically if overwrite is <32KB)
- Majority of files are 8KB or less
- Savings less than the amount of deduplication metadata
- Highly latency sensitive applications (validate performance with testing to see if acceptable or not)

6.3   When to Use Inline Compression or Postprocess Compression Only

Inline compression provides immediate space savings; postprocess compression first writes the blocks to disk as uncompressed and then at a scheduled time compresses the data. Postprocess compression is useful for environments that want compression savings but don't want to incur any performance penalty associated with new writes. Inline compression is useful for customers who aren't as performance sensitive and can handle some impact on new write performance as well as CPU during peak hours. Some considerations for inline and postprocess compression include the following.

Table 3) Considerations for when to use postprocess compression only or also inline compression.

Goal	Recommendation
Minimize Snapshot space	Inline compression: Inline compression will minimize the amount of space used by Snapshot copies.
Minimize disk I/O	Inline compression: Inline compression can reduce the number of new blocks written to disk. Postprocess compression requires an initial write of all the uncompressed data blocks followed by a read of the uncompressed data and a new write of compressed blocks.
Do not affect performance when writing new data to disk	Postprocess compression: Postprocess compression writes the new data to disk as uncompressed without any impact on the initial write performance. You can then schedule when compression occurs to gain the savings.
Minimize effect on CPU during peak hours	Postprocess compression: Postprocess compression allows you to schedule when compression occurs, thus minimizing the impact of compressing new data during peak hours.

Note: It is important to determine that you have sufficient resources available on your system before considering enabling inline compression including during peak hours. NetApp highly recommends that the performance impact be fully tested and understood before you implement in production.

7   Space Savings

This section discusses the potential storage savings for three scenarios: deduplication only, inline compression only (disabling the postprocess schedule), and the combination of compression and deduplication.

Comprehensive testing with various datasets was performed to determine typical space savings in different environments. These results were obtained from various customer deployments and lab testing, and depend upon the customer-specific configuration.

Table 4) Typical deduplication and compression space savings.

Dataset Type	Application Type	Inline Compression Only	Deduplication Only	Deduplication + Compression
File Services/IT Infrastructure		50%	30%	65%
Virtual Servers and Desktops (Boot Volumes)		55%	70%	70%
Database	Oracle OLTP	65%	0%	65%
	Oracle DW	70%	15%	70%
E-Mail	Exchange 2003/2007	35%	3%	35%
	Exchange 2010	35%	15%	40%
Engineering Data		55%	30%	75%
Geoseismic		40%	3%	40%
Archival Data		Archive Application Dependent	25%	Archive Application Dependent
Backup Data		Backup Application Dependent	95%	Backup Application Dependent

These results are based on internal testing and customer feedback, and they are considered realistic and typically achievable. Savings estimates can be validated for an existing environment by using the Space Savings Estimation Tool, as discussed below.

Note: The deduplication space savings in Table 4 result from deduplicating a dataset one time, with the following exception. In cases in which the data is being backed up or archived over and over again, the realized storage savings get better and better, achieving 20:1 (95%) in many instances. The backup case also assumes that the backup application is maintaining data coherency with the original, and that the data's block alignment will not be changed during the backup process. If these criteria are not true, then there can be a discrepancy between the space savings recognized on the primary and secondary systems.

In the NetApp implementation, compression is run before deduplication. This provides us with the ability to use inline compression to get immediate space savings from compression followed by additional savings from deduplication. In our testing of other solutions we found that better savings were achieved by running compression prior to deduplication.

7.1 Factors That Affect Savings

Type of Data

Some nonrepeating archival data such as image files and encrypted data is not considered a good candidate for deduplication.

Data that is already compressed by a hardware appliance or an application, including a backup or an archive application, and encrypted data are generally not considered good candidates for compression.

Space Savings of Existing Data

A major benefit of deduplication and data compression is that they can be used to compress and deduplicate existing data in the FlexVol volumes and data constituents. It is realistic to assume that there will be Snapshot copies—perhaps many—of the existing data.

When you first run deduplication on a flexible volume, the storage savings will probably be rather small or even nonexistent. Although deduplication has processed the data within the volume, including data within Snapshot copies, the Snapshot copies will continue to maintain locks on the original duplicate data. As previous Snapshot copies expire, deduplication savings will be realized. The amount of savings that will be realized when the Snapshot copies expire will depend on the amount of duplicates that were removed by deduplication. For example, consider a volume that contains duplicate data and that the data is not being changed, to keep this example simple. Also assume that there are 10 Snapshot copies of the data in existence before deduplication is run. If deduplication is run on this existing data there will be no savings when deduplication completes, because the 10 Snapshot copies will maintain their locks on the freed duplicate blocks. Now consider deleting a single Snapshot copy. Because the other 9 Snapshot copies are still maintaining their lock on the data, there will still be no deduplication savings. However, when all 10 Snapshot copies have been removed, all the deduplication savings will be realized at once, which could result in significant savings.

During this period of old Snapshot copies expiring, it is fair to assume that new data is being created on the flexible volume and that Snapshot copies are being created. The storage savings will depend on the amount of deduplication savings, the number of Snapshot copies, and when the Snapshot copies are taken relative to deduplication.

Therefore the question is when to run deduplication again in order to achieve maximum capacity savings. The answer is that deduplication should be run, and allowed to complete, before the creation of each and every Snapshot copy; this provides the greatest storage savings benefit. However, depending on the flexible volume and data constituent size and the possible performance impact on the system, this may not always be advisable.

When you run compression against the existing data with the `-shared-blocks true` or `-snapshot-blocks true` option, the system may temporarily show increased space usage. The `-snapshot-blocks true` option compresses blocks that are locked in a Snapshot copy. This may cause new blocks to be written that contain compressed data while the original uncompressed blocks are still locked in a Snapshot copy. When the Snapshot copy expires or is deleted, the savings are realized. When the `-shared-blocks true` option is used, it rewrites the previously shared blocks. This can temporarily take up additional space because the deduplication savings are temporarily lost. When postprocess compression of the existing data is finished, you should rerun deduplication to regain the deduplication savings. This will happen automatically after compression completes by using the command `volume efficiency start –scan-old-data true -compression true -dedupe true –shared-blocks true –snapshot-blocks true`.

Deduplication Metadata

Although deduplication can provide substantial storage savings in many environments, a small amount of storage overhead is associated with it. This should be considered when sizing the FlexVol volume or data constituent. For more information see the Deduplication Metadata Overhead section, above.

Snapshot Copies

Snapshot copies will affect your savings from both deduplication and compression. Snapshot copies lock data, and thus the savings are not realized until the lock is freed by either the Snapshot copy expiring or being deleted. This will be more prevalent in compression-enabled volumes if you perform a significant number of small overwrites. For more information on Snapshot effects on savings for both compression and deduplication, see the Compression and Deduplication with Snapshot Copies section later in this document.

Data That Will Not Compress or Deduplicate

Deduplication metadata (fingerprint file and change logs) is not compressed or deduplicated. Other metadata, such as directory metadata, is also not deduplicated nor compressed. Therefore, space savings may be low for heavily replicated directory environments with a large number of small files (for example, Web space). Information in the Infinite Volume namespace constituent will not compress or deduplicate.

Backup of the deduplicated/compressed volume using NDMP is supported, but there is no space optimization when the data is written to tape because it's a logical operation. (This could actually be considered an advantage, because in this case the tape does not contain a proprietary format.) To preserve the deduplication/compression space savings on tape, NetApp recommends NetApp SMTape.

Only data in the active file system will yield compression/deduplication savings. Data pointed to by Snapshot copies that were created before deduplication processes were run is not released until the Snapshot copy is deleted or expires. Postprocess compression that runs on a schedule will always compress data even if it is locked in a Snapshot copy. Data pointed to by Snapshot copies that were created before starting compression of existing data is bypassed unless using the `-snapshot-blocks true` option. For more information about deduplication/compression and Snapshot copies, refer to the Snapshot Copies section, below.

7.2   Space Savings Estimation Tool (SSET)

The actual amount of data space reduction depends on the type of data. For this reason, the Space Savings Estimation Tool (SSET 3.0) should be used to analyze the actual dataset to determine the effectiveness of deduplication and data compression. SSET performs nonintrusive testing of a dataset. When executed, SSET crawls through all the files in the specified path and estimates the space savings that will be achieved by deduplication only or deduplication and compression. Although actual deduplication and compression space savings may deviate from what the estimation tool predicts, use and testing so far indicate that, in general, the actual results are within +/–5% of the space savings that the tool predicts. The Space Savings Estimation Tool is available for Linux® and Microsoft® Windows® systems, which have the data available locally or use CIFS/NFS; the data does not need to reside on a NetApp storage system for SSET to perform an analysis.

It is limited to evaluating a maximum of 2TB of data. If the given path contains more than 2TB, the tool processes the first 2TB of data, indicates that the maximum size has been reached, and displays the results for the 2TB of data that it processed. The rest of the data is ignored.

For more information about SSET, see the SSET readme file. The SSET tool, including the readme file, can be downloaded from the NetApp communities Web site.

8   Performance

This section discusses the performance aspects of data compression and deduplication.

Since compression and deduplication are part of Data ONTAP, they are tightly integrated with the NetApp WAFL® (Write Anywhere File Layout) file structure. Because of this, compression and deduplication are optimized to perform with high efficiency. They are able to leverage the internal characteristics of Data ONTAP to perform compression and uncompression, create and compare digital fingerprints, redirect data pointers, and free up redundant data areas.

However, the following factors can affect the performance of the compression and deduplication processes and the I/O performance of compressed/deduplicated volumes.

- The application and the type of dataset being used
- The data access pattern (for example, sequential versus random access, the size of the I/O)
- The amount of duplicate data
- The compressibility of the data
- The amount of total data
- The average file size
- The nature of the data layout in the volume
- The amount of changed data between compression/deduplication runs
- The number of concurrent compression/deduplication processes running
- The number of volumes or data constituents that have compression/deduplication enabled
- The hardware platform—the amount of CPU in the system
- The amount of load on the system
- Disk types ATA/SAS, and the RPM of the disk
- The number of disk spindles in the aggregate
- The priority set for the volume efficiency operation assigned to the volume (available starting in clustered Data ONTAP 8.2)

Compression and deduplication can be scheduled to run during nonpeak hours. This allows the bulk of the overhead on the system during nonpeak hours. When there is a lot of activity on the system, compression/deduplication runs as a background process and limits its resource usage. When there is not a lot of activity on the system, compression/deduplication speed will increase, and it could utilize all available system resources. The potential performance impact should be fully tested prior to implementation.

Since compression/deduplication is run on a per FlexVol volume or per-data constituent basis, the more data constituents or FlexVol volumes per node you have running in parallel, the greater the impact on system resources. NetApp recommends for compression/deduplication that you stagger the volume efficiency schedule for volumes to help control the overhead.

Starting in clustered Data ONTAP 8.2, storage QoS provides users the choice of running a volume efficiency operation with a priority of either "best effort" or "background". This allows the administrator to the ability to define how volume efficiency operations compete for resources with user workloads and other system processes not running in background.

When considering adding compression or deduplication, remember to use standard sizing and testing methods that would be used when considering the addition of applications to the storage system. It is important to understand how inline compression will affect your system, how long postprocess operations will take in your environment, and whether you have the bandwidth to run these with acceptable impact on the applications running on your storage system.

Although we have optimized compression to minimize impact on your throughput, there may still be an impact even if you are only using postprocess compression, since we still have to uncompress some data in memory when servicing reads. This impact will continue so long as the data is compressed on disk regardless of whether compression is disabled on the volume at a future point. See the section on uncompression in this document for more details.

Because of these factors, NetApp recommends that performance with compression/deduplication be carefully measured in a test setup and taken into sizing consideration before deploying compression/deduplication in performance-sensitive solutions.

8.1   Performance of Compression and Deduplication Operations

The performance of postprocess compression and deduplication processes varies widely depending on the factors previously described, and this determines how long it takes this postprocess operation to finish running.

Starting in clustered Data ONTAP 8.2, storage QoS provides the administrator the choice of running volume efficiency operations on a volume with a priority of either "best effort" or "background". Running a volume efficiency operation with the default priority "best-effort" allows deduplication and compression to complete the most quickly; however, there might be some impact on the performance of client I/O on the system. Running a volume efficiency operation with a priority of "background" might take longer to complete but should have less impact on client I/O. See the Volume Efficiency Priority section for more details.

Some examples of deduplication and compression performance on a FAS6080 with no other load are listed in Table 5. These values show the sample compression and deduplication performance for a single process and for several parallel processes running concurrently.

Table 5) Postprocess compression and deduplication sample performance on FAS6080.

Number of Concurrent Processes	Compression	Deduplication
1	140MB/s	150MB/s
8	210MB/s	700MB/s

Note: These values indicate potential performance on the listed systems. Your throughput may vary depending on the factors previously described.

The total bandwidth for multiple parallel compression/deduplication processes is divided across the multiple sessions, and each session gets a fraction of the aggregated throughput.

To get an idea of how long it takes for a single deduplication process to complete, suppose that the deduplication process is running on a flexible volume at a conservative rate of 100MB/sec on a FAS3140. If 1TB of new data was added to the volume since the last deduplication update, this deduplication operation takes about 2.5 to 3 hours to complete. Remember, other factors such as different amounts of duplicate data or other applications running on the system can affect the deduplication performance.

To get an idea of how long it takes for a single compression process to complete, suppose that the compression process is running on a flexible volume that contains data that is 50% compressible and at a conservative rate of 70MB/sec on a FAS6080. If 1TB of new data was added to the volume since the last compression process ran, this compression operation takes about 4 hours to complete. Remember, other factors such as different amounts of compressible data, different types of systems, and other applications running on the system can affect the compression performance.

These scenarios are merely examples. Deduplication typically completes much faster following the initial scan, when it is run nightly. Running compression and deduplication nightly can minimize the amount of new data to be compressed / deduplicated, requiring less time to complete.

Inline Compression Performance

Inline compression will consume extra CPU resources whenever data is read or written to the volume; this includes peak hours. The more FlexVol volumes or data constituents per node that are enabled with compression, the more the resource demand and overhead will be. The impact will be shown by longer latencies on the volume that has compression enabled. Given the possible impact on peak time performance, NetApp recommends limiting typical use cases to those not as performance sensitive, such as file services, backup, and archive solutions.

One of the biggest impacts from inline compression occurs when attempting compression on files with little or no savings. To offset this impact, starting in Data ONTAP 8.2 we added the idd option (incompressible data detection) to detect and skip inline compressing of these files or compression groups.

The way it works depends on whether you are writing a large or small file. By default we assume a file that is less than 500MB is a small file. You can modify the size of what is considered a small or large file by modifying the `quick-check-fsize` option of the `volume efficiency config` command.

For small files we compress each compression group within the file unless or until we find one that has less than 25% savings. At that time we mark the file as incompressible and stop any additional inline compress attempts. For large files we try to compress the first 4K of each compression group. If there is at least 25% savings in the first 4K, we continue compressing the rest of the compression group. If the first 4K of a compression group does not have at least 25% savings, we do not continue inline compression attempts on the rest of the compression group and write it as uncompressed. We continue to do this quick check on all compression groups within the file.

Postprocess compression will continue to try to compress any compression groups that were skipped by inline compression.

This low compressibility check is optional and can be enabled using the `volume efficiency` command with the `-idd true|false` option. This option is enabled|disabled on a volume-by-volume basis.

To determine if you should consider enabling this option, look at the output of the `volume efficiency stat` command, specifically the following two counters:

```
Inline Compression Attempts :  Total number of inline compression attempted
Inline Incompressible CGs   :  Total number of incompressible CG found by
inline compression
```

This should be checked over time to determine if the data has a high ratio of incompressible compression groups (CGs). For example, an incompressible CG to inline compression attempt ratio of =>0.8 over a period of time would show that you should consider setting this option to true. By default, this option is set to false.

While compression requires deduplication to be enabled on your volume, you can choose to run compression alone. To do this you would enable both deduplication and compression (both postprocess and inline) and set the schedule for postprocess compression and deduplication to never run. Although this might not give the best space savings it would be valuable for use cases that benefit from compression savings but do not deduplicate well and do not have a window in which they want to incur the resource overhead associated with postprocess operations.

It is important that you have sufficient resources available on your system during peak hours before considering enabling inline compression. NetApp highly recommends that the performance impact be fully tested and understood before implementing the process in production.

Considerations

The more compressible the data, the faster compression occurs. In other words it will be faster to compress data that has 75% savings from compression compared to compressing data that has only 25% savings from compression.

Deduplication throughput may decrease on a volume that contains compressed data, depending on the amount of sharing and the compressibility of the data.

Experience has shown that the more new data that is written to the FlexVol volume or data constituent, relative to the existing amount of logical data, the better the performance of the deduplication process for that FlexVol volume or data constituent. The amount of sharing per block will also affect performance.

These factors further enforce the strong recommendation for performance testing with compression and deduplication prior to implementation.

8.2 Impact on the System During Compression and Deduplication Processes

Both compression and deduplication are processes that by design will use all available CPU cycles that other applications are not using. However, they can still affect the performance of other applications running on the system.

The number of compression and deduplication processes that are running and the phase that the deduplication process is running in can affect the performance of other applications running on the system. Up to eight concurrent compression/deduplication scans can run concurrently on the same NetApp storage system. If there is an attempt to run an additional compression/deduplication process beyond the maximum, the process will be placed in a queue and automatically started when there are free processes.

Here are some observations about running deduplication on a FAS3140 system:

- With eight deduplication processes running and no other processes running, deduplication uses 15% of the CPU in its least invasive phase. By design it will use nearly all of the available CPU in its most invasive phase unless a higher-priority request comes in.

- With eight compression processes running and no other processes running, by design compression will use all available CPU unless a higher-priority request comes in, such as from an application.

- When one deduplication process is running, there is 0% to 15% performance degradation on other applications.

- With eight deduplication processes running, there may be a 15% to more than a 50% performance penalty on other applications running on the system.

8.3 Impact on the System from Inline Compression

Enabling compression on a system increases CPU utilization. As mentioned above, the way compression affects your system depends on a number of variables. On workloads such as file services, systems with less than 50% CPU utilization have shown an increased CPU usage of ~20% for datasets that were 50% compressible. For systems with more than 50% CPU utilization, the impact may be more significant. The impact on your environment will vary depending on a number of factors, including those described at the beginning of this section. NetApp recommends testing in a lab environment to fully understand the impact on your environment before implementing into production.

8.4 I/O Performance of Deduplicated Volumes

Write Performance of a Deduplicated Volume

The impact of deduplication on the write performance of a system is a function of the hardware platform that is being used, as well as the amount of load that is placed on the system. For deduplicated volumes, if the load on a system is low—for instance, systems in which the CPU utilization is around 50% or lower—there is a small to negligible difference in performance when writing data to a deduplicated volume; there is no noticeable impact on other applications running on the system. On heavily used systems in which the system is CPU-bound, the impact on write performance may be noticeable. For example, in an extreme case with 100% random overwrites with over 95% savings, a FAS3140 showed a performance impact of 15%. On high-end systems such as the FAS6080 system, the same scenario showed a performance impact of 15–30% for random writes. The impact was lower with multiple volumes or data constituents. NetApp highly recommends Flash Cache™ or Flash Pool™ intelligent caching for metadata caching in heavy write scenarios. The performance impact of sequential writes, such as new files or appends, was less than 5% as compared to volumes without deduplication enabled.

Note: The deduplication numbers are for FC drives. If SATA drives are used in a system, the performance impact may be greater.

The performance impact will vary and should be tested before implementing in production.

Read Performance of a Deduplicated Volume

When data is read from a deduplication-enabled volume, the impact on the read performance varies depending on the difference between the deduplicated block layout and the original block layout. There is minimal impact on random reads. Data ONTAP has specific optimizations, referred to as *intelligent cache,* that reduce the performance impact that deduplication has on sequential read workloads. Because deduplication alters the data layout on the disk, using deduplication without intelligent cache could affect the performance of sequential read applications such as dump source and other sequential read-heavy applications. In addition, the Flash Cache cards as well as Flash Pool also utilize intelligent caching to optimize performance, and should be highly considered when using deduplication.

In scenarios in which deduplication savings are lower, deduplication has little or no performance impact on sequential reads. In test scenarios in which there were high amounts of deduplication savings, say 100%, there was a throughput enhancement of 50%; in the worst-case scenarios, in which intelligent cache was bypassed by forcing sequential reads of noncached blocks, there was performance degradation of up to 25% on a CPU-bound system. Having at least 15% CPU available and 10% disk I/O availability (disk busy <90%) will typically absorb the impact deduplication would have on sequential reads. Typical real-world scenarios will be somewhere between the two extreme test cases described above, and should be tested. Intelligent cache has proven to be very useful in many scenarios, especially in virtualized environments.

Workload Impact While Deduplication Process Is Active

A differentiator of NetApp deduplication is the fact that it runs as a postprocess, allowing the storage systems to run with little or no impact from deduplication during critical production hours. The deduplication process can be postponed until a more appropriate time when resources are more readily available on the storage system. When the background deduplication process runs as scheduled or triggered, it searches for duplicates and updates the file system to remove the duplicates. This process should be tested to understand the impact on your systems, and scheduled appropriately. During this process, deduplication will use system resources and host access will typically see a performance impact of 20–30% on random writes. The random reads are more sensitive to the different phases of the deduplication process, and can see a performance impact of 15–70% while the deduplication process is running. Starting in clustered Data ONTAP 8.2 you can minimize this impact by using a priority of background on the volume efficiency operations.

8.5 I/O Performance of Compressed Volumes

Compression has an impact on I/O performance. File services–type benchmark testing with compression savings of 50% has shown a decrease in throughput of ~5%. The impact on your environment will vary depending on a number of factors, including the amount of savings, the type of storage system, how busy your system is, and other factors laid out at the beginning of this section. NetApp highly recommends testing in a lab environment to fully understand the impact on your environment before implementing in production.

Write Performance of a Compressed Volume

The impact of compression on the write performance of a system is different depending on whether you are using inline or postprocess compression.

If you use inline compression, the write performance is a function of the hardware platform that is being used, the type of write (that is, partial or full), the compressibility of the data, the number of FlexVol volumes or data constituents per node with compression enabled, as well as the amount of load that is placed on the system.

For postprocess compression the write performance will only be impacted for partial overwrites of previously compressed data; all other data will be written uncompressed. It will be compressed the next time postprocess compression is run.

For disaster recovery environments such as volume SnapMirror with datasets that provide good space savings, there is no CPU impact and there is reduced I/O on the destination system, faster replications, as well as network bandwidth savings during the transfer. For more information on volume SnapMirror and SnapVault with data compression, refer to the section on Feature Interoperability, below.

Read Performance of a Compressed Volume

When data is read from a compressed volume, the impact on the read performance varies depending on the access patterns, the amount of compression savings on disk, and how busy the system resources are (CPU and disk). In a sample test with a 50% CPU load on the system, read throughput from a dataset with 50% compressibility showed decreased throughput of 25%. On a typical system the impact could be higher because of the additional load on the system. Typically the most impact is seen on small random reads of highly compressible data, and on a system that is more than 50% CPU busy. Impact on performance will vary and should be tested before implementing in production.

8.6 Flash Cache Cards

In environments with high amounts of shared blocks that are read repeatedly, Flash Cache can significantly reduce the number of disk reads, thus improving the read performance. Flash Cache does not increase performance of the deduplication or compression operations. Flash Cache cards do not cache sequential reads; therefore they don't cache compressed blocks on disk. The amount of performance improvement from Flash Cache depends on the amount of shared blocks, the access rate, the active dataset size, and the data layout.

Flash Cache has provided significant performance improvements in VMware® VDI environments. These advantages are further enhanced when combined with shared block technologies, such as NetApp deduplication or NetApp FlexClone® technology. For more information about the Flash Cache cards in VMware VDI environments, refer to TR-3705: NetApp and VMware VDI Best Practices.

8.7 Flash Pool

In environments with high amounts of shared blocks that are read repeatedly or written randomly, Flash Pool can significantly reduce the number of disk reads and partial overwrites, thus improving performance. Flash Pool does not increase performance of the deduplication or compression operations. Flash Pool does not cache sequential I/O; therefore it doesn't cache compressed blocks on disk. The amount of performance improvement from Flash Pool depends on the amount of shared blocks, the access rate, the active dataset size, and the data layout. For more information on Flash Pool, refer to TR-4070: NetApp Flash Pool Design and Implementation Guide.

9   Considerations for Adding Compression or Deduplication

It is extremely important that you test out any new technology before implementing it into production. Compression and deduplication can have an impact on your performance both during the compression and block-sharing process and after your data has been compressed/deduplicated. Inline compression can have an impact on backup/restore windows, on backup/archive storage, as well as on performance during peak hours on production storage.

NetApp recommends that if testing proves the savings benefit of running compression/deduplication in your environment and the performance impact is acceptable, you should implement one volume at a time per storage system. You should record statistics before enabling the technology to record a baseline. You should further record the statistics after adding the technology to a single volume and after adding any additional volumes. This will help you to understand and better predict the cost of implementation. It will also help to prevent you from exceeding the acceptable performance overhead of the overall storage system.

If you are using compression and deduplication on a backup system, you might consider compressing and deduplicating the existing backup data and using inline compression on the incremental backups. If you're considering running the compression or deduplication on existing data on the source system, you should be aware of the implications. When compression or deduplication on existing data is complete, all newly compressed blocks are written as new blocks and all deduplicated blocks are considered changed blocks. This can lead to the next incremental transfer being unexpectedly large. For more information, refer to the volume SnapMirror section later in this document.

You should not enable deduplication or compression on volumes that do not have savings in excess of the amount of deduplication metadata. This is true for compression as well since compression requires deduplication to be enabled. Refer to the section on deduplication metadata overhead for details.

Additional examples of when not to enable deduplication include applications that overwrite data at a rapid rate, and applications that perform small writes and add unique headers. An example of this would be an Oracle Database that is configured with an 8KB block size.

Additional examples of when not to use compression on a volume include if the workload has a high change rate that consists of a large number of partial file overwrites that are <32K; this would be further exacerbated if there is long Snapshot copy retention. Other examples are data that is already compressed by applications, files that are <=8K, and when the application is highly latency sensitive (validate performance with testing to see if it is acceptable or not).

Because we attempt to compress all files in a compression-enabled volume, the performance impact will be there regardless of whether we get savings from compression on the files or not.

Some use cases don't show enough savings with either deduplication or compression to justify the overhead. Another reason to not run either compression or deduplication is if the system can't afford any additional overhead at any time. A couple of examples of datasets that don't show savings are rich media files, encrypted data, and video surveillance. For more information on how to assess your system for using deduplication, refer to TR-3936: Playbook: Easily Assess Your Environment for NetApp Deduplication.

9.1 Microsoft SQL Server

A Microsoft SQL Server database will use 8KB page sizes. Although Microsoft SQL Server will place a unique header at the beginning of each page, the rest of the blocks within each page may still contain duplicates. This means that deduplication may provide significant savings when comparing the 4KB blocks within the volume, compression may also provide savings. The Space Savings Estimation Tool (SSET 3.0) can be used to estimate the amount of savings that would be achieved with compression or deduplication or both.

9.2 Microsoft SharePoint

Compression and deduplication can be used together and are transparent to Microsoft SharePoint®. Block-level changes are not recognized by SharePoint, so the SharePoint database remains unchanged in size, even though there are capacity savings at the volume level.

9.3 Microsoft Exchange Server

If Microsoft Exchange and NetApp data compression or deduplication will be used together, consider the following points.

- In some Exchange environments, extents are enabled to improve the performance of database validation. Enabling extents does not rearrange blocks on disk that are shared between files by deduplication on deduplicated volumes. Enabling extents does not predictably optimize sequential data block layout when used on deduplicated volumes, so there is no reason to enable extents on deduplicated volumes.

- NetApp data compression shows space savings in the range of 35% for all versions of Microsoft Exchange. NetApp recommends running the SSET on your environment to better estimate the compression savings that your environment can achieve.

- Beginning with Microsoft Exchange 2010, single-instancing storage will no longer be available. NetApp deduplication for FAS and V-Series provides significant savings for primary storage running Exchange 2010.

For additional details about Exchange, refer to TR-3578: Microsoft Exchange Server 2007 Best Practices Guide or TR-3824: Storage Efficiency and Best Practices for Microsoft Exchange Server 2010.

9.4 Lotus Domino

The compression and deduplication space savings that you can expect will vary widely with the type (e-mail, applications, and so on) and frequency of data in your environment. NetApp customers using Domino have reported anywhere from 8% to 60% deduplication savings in their Domino environment. NetApp recommends running the SSET tool on your environment to better estimate the compression and deduplication savings that could be achieved on your environment.

Domino 8.5 introduced a feature called Domino Attachment and Object Service (DAOS). NetApp deduplication will still be effective when DAOS is enabled, but NetApp anticipates that the reported space savings will be lower since DAOS has already performed much of the work.

If Domino database encryption is enabled for all or the majority of databases, you should anticipate that both deduplication and compression space savings will be very small. This is because encrypted data is by its nature unique.

Domino quotas are not affected by deduplication or compression. A mailbox with a limit of 1GB cannot store more than 1GB of data in a deduplicated/compressed volume even if the data consumes less than 1GB of physical space on the storage system.

For additional details about Lotus Domino and deduplication, including advanced configurations to increase the amount of storage savings when using Domino clustered servers with Domino DAOS and NetApp deduplication, refer to TR-3843: Storage Savings with Domino and NetApp Deduplication.

9.5 VMware

VMware environments deduplicate extremely well. However, while working out the VMDK and datastore layouts, keep the following points in mind.

Operating system VMDKs deduplicate extremely well because the binary files, patches, and drivers are highly redundant between virtual machines (VMs). Maximum savings can be achieved by keeping these in the same volume. These VMDKs typically do not benefit from compression over what deduplication can already achieve. Further, since compressed blocks bypass the Flash Cache card, compressing the operating system VMDK can negatively impact the performance during a boot storm. For these reasons, NetApp does not recommend adding compression to an operating system VMDK. See the Flash Cache Interoperability section, in the Feature Interoperability below, for more details.

Application binary VMDKs compress/deduplicate to varying degrees. Duplicate applications deduplicate very well, applications from the same vendor commonly have similar libraries installed and deduplicate somewhat successfully, and applications written by different vendors don't deduplicate at all.

When compressed/deduplicated, application datasets have varying levels of space savings and performance impact based on application and intended use. Careful consideration is needed, just as with nonvirtualized environments, before deciding to keep the application data in a compressed/deduplicated volume.

Transient and temporary data such as VM swap files, page files, and user and system temp directories do not compress or deduplicate well and potentially add significant performance pressure when compressed/deduplicated. Therefore NetApp recommends keeping this data on a separate VMDK and volume that are not compressed/deduplicated. For more information on page files refer to TR-3749: NetApp and VMware vSphere Storage Best Practices.

NetApp includes a performance enhancement referred to as *intelligent cache*. Although it is applicable to many different environments, intelligent caching is particularly applicable to VM environments, where multiple blocks are set to zero as a result of system initialization. These zero blocks are all recognized as duplicates and are deduplicated very efficiently. The warm cache extension enhancement provides increased sequential read performance for such environments, where there are very large amounts of deduplicated blocks. Examples of sequential read applications that benefit from this performance enhancement include NDMP and some NFS-based applications. This performance enhancement is also beneficial to the boot-up processes in VDI environments.

The expectation is that about 30% space savings will be achieved overall. This is a conservative figure, and in some cases users have achieved savings of up to 80%. The major factor that affects this percentage is the amount of application data. New installations typically deduplicate extremely well, because they do not contain a significant amount of application data.

Important: In VMware environments, the need for proper partitioning and alignment of the VMDKs is extremely important (not just for deduplication). VMware must be configured so that the VMDKs are aligned on NetApp WAFL (Write Anywhere File Layout) 4K block boundaries as part of a standard VMware implementation. To learn how to prevent the negative performance impact of LUN/VMDK misalignment, read TR-3747: Best Practices for File System Alignment in Virtual Environments, TR-3428: NetApp and VMware Best Practices Guide, or TR-3749: NetApp and VMware vSphere Storage Best Practices. Also note that the applications in which performance is heavily affected by deduplication (when these applications are run *without* VMware) are likely to suffer the same performance impact from deduplication when they are run *with* VMware.

For more information about NetApp storage in a VMware environment, see TR-3428: NetApp and VMware Virtual Infrastructure 3 Storage Best Practices.

9.6 Oracle

Data compression and deduplication can provide significant savings in Oracle environments, but proper testing should be done to determine the savings for your environment. The Space Savings Estimation Tool (SSET 3.0) can be used to estimate the amount of savings that would be achieved with deduplication or compression or both.

Data compression has shown savings of 60–75% in customer environments. Deduplication savings depend on the Oracle configuration.

A typical Oracle data warehouse or data mining database will typically use 16KB or 32KB page sizes. Although Oracle will place a unique identifier at the beginning of each page, the rest of the blocks within each page may still contain duplicates. This means that deduplication may provide significant savings when comparing the 4KB blocks within the volume.

Oracle OLTP databases typically use an 8KB page size. Oracle will once again place a unique identifier at the beginning of each page. Testing has shown that these environments typically do not have a significant amount of duplicate blocks and do not show very much deduplication savings.

One additional case to consider is if a table space is created or extended. In this case Oracle will initialize the blocks and commit many of them in the same transaction. This will result in the creation of duplicate blocks, allowing deduplication to provide savings. However, as these blocks are filled with incoming data the space savings will shrink.

9.7 Symantec Backup Exec

Deduplication savings with Symantec™ Backup Exec™ will not be optimal because Backup Exec does not block-align data when it writes files out to its volumes. The net result is that there are fewer duplicate blocks available to deduplicate.

9.8 Backup

There are various ways to back up your data. Here are a few compression/deduplication considerations for your backup environment.

- To achieve optimal backup throughput it is a best practice to:
  - Determine that postprocess compression/deduplication operations initiate only after your backup completes.
  - Postprocess compression and deduplication operations on the destination volume complete prior to initiating the next backup.
- When backing up data from multiple volumes to a single volume you may achieve significant space savings from deduplication beyond that of the deduplication savings from the source volumes. This is because you are able to run deduplication on the destination volume, which could contain duplicate data from multiple-source volumes.
- If you are backing up data from your backup disk to tape, consider using SMTape to preserve the deduplication/compression savings. Utilizing NDMP to tape will not preserve the deduplication/compression savings on tape.
- If you are using compression and deduplication on a backup system, you might consider compressing and deduplicating the existing backup data and using inline compression on the incremental backups.
- NetApp recommends using inline data compression on your backup volumes so long as you are not resource constrained on your storage system. If your system has a CPU bottleneck during backups, NetApp recommends either reducing the number of concurrent backups or only using postprocess compression instead of inline.
- If the application that you are using to perform backups already does compression, NetApp data compression will not add significant additional savings.

9.9   Tivoli Storage Manager

If IBM Tivoli Storage Manager (TSM) and NetApp deduplication or compression will be used together, consider the following points.

- Deduplication savings with TSM will not be optimal because TSM does not block-align data when it writes files out to its volumes. The net result is that there are fewer duplicate blocks available to deduplicate.
- TSM compresses files backed up from clients to preserve bandwidth. On data that is compressed using Tivoli Storage Manager, compression does not usually yield much additional savings when deduplicated or compressed by the NetApp storage system.
- TSM client-based encryption results in data with no duplicates. Encrypted data does not usually yield good savings when deduplicated or compressed.
- TSM's progressive backup methodology backs up only new or changed files, which reduces the number of duplicates, since there are not multiple full backups to consider.

10 Best Practices for Optimal Savings and Minimal Performance Overhead

This section describes best practices and lessons learned from internal tests and customer deployments.

- Both deduplication and compression consume system resources and can alter the data layout on disk. Due to the application's I/O pattern and the effect of deduplication/compression on the data layout, the read and write I/O performance can vary. The space savings and the performance impact depend on the application and the data contents.
- NetApp recommends that the performance impact of deduplication/compression be carefully considered and measured in a test setup and taken into sizing consideration before deploying deduplication/compression in performance-sensitive solutions. For information about the impact of deduplication/compression on other applications, contact the specialists at NetApp for their advice and test results for your particular application.
- If there is only a small amount of new data, run deduplication infrequently, because there is no benefit in running it frequently in such a case, and it consumes system resources. The frequency for running deduplication depends on the rate of change of the data in the volume.
- The more concurrent compression/deduplication processes you run on a system, the more system resources are consumed.
- If NetApp Snapshot copies are required, run the compression/deduplication processes before creating the Snapshot copies to minimize the amount of data that gets locked in to the copies. (Make sure that the compression/deduplication processes have completed before creating the Snapshot copy.) If a Snapshot copy is created on a volume before the deduplication processes have completed, the result is likely to be lower space savings. If a Snapshot copy is created on a volume before the compression processes have completed, the result is likely to be that more space is used by the Snapshot copies.
- For deduplication to run properly, you need to leave some free space for the deduplication metadata. For information about how much extra space to leave in the FlexVol volume, data constituent, and aggregate, see the section on Deduplication Metadata Overhead, above.

11 Configuration and Operation

This section discusses how to configure and interpret savings, as well as provides end-to-end examples of deduplication and/or compression. Although the section discusses some basic things, we assume that the NetApp storage system is already installed and running and that you are familiar with basic NetApp administration.

11.1 Command Summary

The following sections describe the various compression and deduplication commands. Starting in clustered Data ONTAP 8.1, we have replaced the `sis` commands with new `volume efficiency` commands; however, for backward compatibility both will exist with identical behavior. The new `volume efficiency` commands are enhanced and offer more advanced options.

Compression and Deduplication of New Data Commands

These commands are available for configuring compression and deduplication on new data written to disk after compression/deduplication is enabled. To compress or deduplicate existing data on disk, see the commands in the next table.

Table 6) Commands for compression and deduplication of new data.

Command	Summary
volume efficiency start ? volume efficiency stop ?	Lists the commands for starting and stopping compression.
volume efficiency start –vserver <SVMname> -volume <volname> -qos-policy background	Starts postprocess compression and deduplication on a volume with a priority of background instead of the default of best-effort. Regardless of what priority is set in the assigned volume efficiency policy, if –qos-policy is specified in the manual start, this setting will be used for the manual start. (Available starting in clustered Data ONTAP 8.2.)
volume efficiency on –vserver <SVMname> –volume <volname>	Enables deduplication on the volume specified.
volume efficiency modify –vserver <SVMname> –volume <volname> –compression true	Enables postprocess compression of subsequently created data. This requires that deduplication first be enabled on the volume.
volume efficiency modify –vserver <SVMname> –volume <volname> –compression true –inline-compression true	Enables inline compression and postprocess of subsequently created data. This requires that deduplication first be enabled on the volume. There is no way to enable inline compression without postprocess compression and deduplication.
volume efficiency modify –vserver <SVMname> –volume <volname> -inline-compression false	Disables inline compression. It will not stop postprocess compression or decompress any existing compressed data.
volume efficiency modify –vserver <SVMname> –volume <volname> –idd true|false - `quick-check-fsize <size in bytes>`	The -idd option enables or disables incompressible data detection for inline compression. The –quick-check-fsize option controls what size of file is considered large. This is new in Data ONTAP 8.2 and is off by default.
volume efficiency show –vserver <SVMname> –volume <volname> -fields progress	Checks the progress of postprocess compression/deduplication operations running on a volume. The `volume efficiency show` command does not currently display information about status or progress for an Infinite Volume. To see the status or progress of individual data constituents, substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true –fields progress`.

Command	Summary
volume efficiency show -fields compression, inline-compression	Displays which volumes have inline or postprocess compression enabled.
volume efficiency policy show	Displays the compression/deduplication volume efficiency policies available within the cluster.
volume efficiency policy create –vserver <SVMname> - policy <policyname> - schedule <cron job schedule> –duration <time interval> -enabled true	Creates a volume efficiency policy for a specific Vserver. This policy can then be assigned to a volume. The value for –schedule must correlate to an existing cron job schedule within the cluster. The value for –duration represents how many hours you want compression/deduplication to run before stopping.
volume efficiency policy modify –vserver <SVMname> –policy <policy_name> - schedule <cron job schedule> -duration <time interval> -enabled true	Modifies an existing compression/deduplication volume efficiency policy for a specific Vserver. This will change the schedule for all volumes that are using the volume efficiency policy.
volume efficiency policy modify –vserver <SVMname> –policy <policy_name> - qos-policy background	Change the priority assigned to a specific volume efficiency policy to use background instead of the default best-effort. (Available starting in clustered Data ONTAP 8.2.)
volume efficiency policy modify –vserver <SVMname> –policy <policy_name> - qos-policy best-effort	Change the priority assigned to a specific volume efficiency policy use best-effort instead of background. (Available starting in clustered Data ONTAP 8.2.)
volume efficiency policy delete –vserver <SVMname> –policy <policy_name>	Deletes an existing compression/deduplication scheduling policy for a specific Vserver.
volume efficiency show -fields policy	Displays the assigned scheduling policies for each volume that has deduplication enabled.
volume efficiency modify –vserver <SVMname> –volume <volname> -policy <policy_name>	Changes which compression/deduplication scheduling policy is assigned to a volume.
volume efficiency modify –vserver <SVMname> –volume <volname> - schedule auto	Modifies the compression/deduplication scheduling policy for a specific volume to run only when 20% new data is written to the volume. The 20% threshold can be adjusted by adding the @num option to auto (auto@num), where num is a two-digit number to specify the percentage.
volume efficiency start –vserver <SVMname> –volume <volname>	Begins the deduplication process on the volume specified. If compression is also enabled it will start first followed by deduplication. This will compress and deduplicate any data that has been written to disk after compression/deduplication was enabled on the volume. This will not compress or deduplicate data that existed on the volume prior to compression/deduplication being enabled. For this, refer to the postprocess compression/deduplication of existing data commands below.
volume efficiency start –vserver <SVMname> –volume <volname> - delete-checkpoint true	Deletes the existing checkpoint information. This option is used to delete checkpoint information that is still considered valid. By default, checkpoint information is considered invalid after 24 hours.
volume efficiency stop –vserver <SVMname> –volume <volname>	Suspends an active postprocess compression/deduplication operation running on the volume without creating a checkpoint.

Command	Summary
volume efficiency stop –all true *	Stops all currently active and queued postprocess compression/deduplication operations running on a system.
volume efficiency stop –all true –vserver <SVMname> *	Stops all currently active or queued postprocess compression/deduplication operations running in a Vserver.
volume efficiency check –vserver <SVMname> –volume <volname>	Verifies and updates the fingerprint database for the volume specified; includes purging stale fingerprints (requires advanced mode).
volume efficiency check –vserver <SVMname> –volume <volname> -delete-checkpoint true	Deletes the existing checkpoint file and starts the check portion of the deduplication operation from the beginning. (requires advanced mode).
volume efficiency show –vserver <SVMname> –volume <volname>	Displays compression- and deduplication-related information on a volume. The `volume efficiency show` command does not currently display information about status or progress for an Infinite Volume. To see the status or progress of individual data constituents, substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true.`
volume efficiency show –vserver <SVMname> –volume <volname> –fields state	Shows whether or not deduplication is enabled on the volume.
volume efficiency show –vserver <SVMname> –volume <volname> -fields compression	Can be used to check whether compression is enabled on a particular volume.
volume efficiency show –vserver <SVMname> –volume <volname> -fields inline-compression	Can be used to check whether inline compression is enabled on a particular volume.
volume efficiency show –vserver <SVMname> –volume <volname> -fields status	Checks the status of postprocess compression/deduplication operations running on a volume. The `volume efficiency show` command does not currently display information about status or progress for an Infinite Volume. To see the status of individual data constituents, substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true –fields status.`
volume efficiency show -fields schedule	Displays the current compression/deduplication schedule for each volume.
volume efficiency show -fields policy	Displays the current compression/deduplication policies assigned to each volume.
volume show –vserver <SVMname> –volume <volname>	Shows volume details including storage efficiency–related status information.
volume show –vserver <SVMname> –volume <volname> -fields dedupe-space-saved	Shows the total space saved in the volume from deduplication. You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".

Command	Summary
volume show –vserver <SVMname> – volume <volname> -fields dedupe-space-saved-percent	Shows the total percentage savings in the volume from deduplication. (deduplicated /(used + deduplicated) * 100). You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".
volume show –vserver <SVMname> – volume <volname> –fields compression-space-saved	Shows the total space saved in the volume from compression. You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".
volume show –vserver <SVMname> – volume <volname> –fields compression-space-saved-percent	Shows the total percentage savings in the volume from compression (compressed/ (used + compressed) * 100). You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".
volume show –vserver <SVMname> – volume <volname> –fields sis-space-saved	Shows the total space saved in the volume from compression and deduplication. You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".
volume show –vserver <SVMname> – volume <volname> –fields sis-space-saved–percent	Shows the total percentage savings in the volume from compression and deduplication ((compression+deduplication saved)/ (used + compression saved + deduplication saved) * 100). You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".
volume show -fields used, sis-space-saved, sis-space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent, compression-space-saved, compression-space-saved-percent	Shows multiple values combined on one line. This example shows space used, total space saved, percent total savings, deduplication and compression space, and percent savings for all volumes in the cluster. You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true". Can also use df –S to display the same.
df –S <volname>	Shows space savings from compression and deduplication as well as actual physical used capacity per volume.

Compression and Deduplication of Existing Data Commands

These commands are available to compress and deduplicate data that was already on disk prior to compression or deduplication being enabled. This option is typically used upon initial configuration of compression and deduplication on an existing volume that contains uncompressed and undeduplicated data. (There's no need to use this option on a volume that has just been created and doesn't contain any data.)

Table 7) Commands for compression and deduplication of existing data.

Command Data ONTAP 8.1	Summary
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true	Begins compression (if enabled) of existing data followed by deduplication on the volume specified. It will use the latest checkpoint if one exists and is less than 24 hours old. This will bypass compression of any blocks that are already deduplicated or locked in Snapshot copies.
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –use-checkpoint true	Begins compression (if enabled) of existing data followed by deduplication on the volume specified using the existing checkpoint information, regardless of the age of the checkpoint information.
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –delete-checkpoint true	Begins compression (if enabled) of existing data followed by deduplication on the volume specified. It will disregard any checkpoints that exist. This will bypass compression of any blocks that are already deduplicated or locked in Snapshot copies.
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –dedupe true	Begins the deduplication process on the volume specified and performs a scan of the volume to process existing data. It will use the latest checkpoint if one exists. The –dedupe option is used if compression is also enabled on the volume and you only want to run deduplication against the existing data, not compression. The new data will still be compressed (requires advanced mode).
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –dedupe true -delete-checkpoint true	Deletes any previous checkpoints and initiates the deduplication process of existing data to begin from scratch on the volume specified (requires advanced mode).
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –compression true	Begins compression of the existing data on disk. It will use the latest checkpoint if one exists. It will not run deduplication against this data (requires advanced mode).
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –compression true -delete-checkpoint true	Deletes any previous checkpoints and initiates compression of existing data to begin from scratch on a volume. It will not run deduplication against this data (requires advanced mode).
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –compression true -shared-blocks true	Initiates compression of existing data to begin and include shared blocks created by deduplication or cloning of data. The –shared-blocks option can be used together with the –snapshot-blocks option. It will not run deduplication against this data.
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –compression true –snapshot-blocks true	Initiates compression of existing data to begin and include blocks that are locked in existing Snapshot copies. The – snapshot-blocks option can be used together with the – shared-blocks option. It will not run deduplication against this data.
volume efficiency start –vserver <SVMname> –volume <volname> –scan-old-data true –compression true -shared-blocks true –snapshot-blocks true	Initiates compression of all possible blocks containing existing data. It will not run deduplication against this data.
volume efficiency show -fields progress	Checks the progress of postprocess compression/deduplication operations.

Disabling Compression and Deduplication Commands

These commands are available to disable and undo compression and deduplication savings.

Table 8) Commands for disabling / undoing compression and deduplication.

Command Data ONTAP 8.1	Summary
volume efficiency stop ?	Lists the commands for stopping compression and deduplication.
volume efficiency undo ?	Lists the commands for removal of compression and deduplication on data within a volume (requires advanced mode).
volume efficiency stop –vserver <SVMname> –volume <volname>	Suspends an active postprocess compression/deduplication process on the volume without creating a checkpoint.
volume efficiency stop –all –vserver <SVMname> –volume <volname>	Stops both the currently active as well as the queued postprocess compression/deduplication operations on a volume.
volume efficiency modify –vserver <SVMname> –volume <volname> – compression false – inline-compression false	Disables any newly created data from being compressed. It will not decompress any existing compressed data.
volume efficiency off –vserver <SVMname> –volume <volname>	Disables inline and postprocess compression as well as deduplication on the specified volume. This means that there will be no additional change logging, compression, or deduplication operations, but the volume remains a compressed and deduplicated volume, and the storage savings are preserved. If this command is used and then compression or deduplication is turned back on for this volume, the volume should be rescanned with the volume efficiency start –scan-old-data true command to gain the maximum savings.
volume efficiency undo –vserver <SVMname> –volume <volname>	Initiates the removal of compression and block sharing from deduplication and file/LUN cloning on a volume; it first requires that compression and deduplication be turned off on the volume (requires advanced mode).
volume efficiency undo –vserver <SVMname> –volume <volname> - dedupe true	Initiates the removal of block sharing from deduplication and file/LUN cloning on a volume; it first requires that compression and deduplication be turned off on the volume (requires advanced mode).
volume efficiency undo –vserver <SVMname> –volume <volname> - compression true	Initiates the uncompression of data in a volume; it first requires that compression be turned off on the volume (requires advanced mode).
volume efficiency show –vserver <SVMname> –volume <volname>	Returns the current status of the specified volume. Can be used to check the progress of the removal of compression/deduplication running on a particular volume. To see the status or progress of individual data constituents substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true –fields progress`.
volume show -fields used, sis-space-saved, sis-space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent, compression-space-saved, compression-space-saved-percent	Shows multiple values combined on one line. This example shows deduplication and compression space and percent savings for all volumes in the cluster. You can use the same command to display the aggregated savings within an Infinite Volume; or, to see savings at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true".

11.2 Interpreting Space Usage and Savings

The `volume show` command can both show savings as well as actual physical space used per FlexVol volume or Infinite Volume. To see savings within the Infinite Volume at the data constituent level, run the command substituting the "–volume <volname>" option with "-is-constituent true". Here is a summary of the various savings fields that can be displayed with the `volume show` command.

Command:

```
volume show -fields used, sis-space-saved, sis-space-saved-percent, dedupe-
space-saved, dedupe-space-saved-percent, compression-space-saved,
compression-space-saved-percent
```

Table 9) Interpreting `volume show` savings values.

Fields	Description
vserver	Vserver name
volume	Volume name
used	Amount of physical space used on the volume
sis-space-saved	Number of bytes saved by both compression and deduplication
sis-space-saved-percent	Percentage space saved from both compression and deduplication (compression+deduplication bytes saved) / (used + compression bytes saved + deduplication bytes saved) * 100
dedupe-space-saved	Number of bytes saved by deduplication
dedupe-space-saved-percent	Percentage of space saved from deduplication dedupe bytes saved / (used + dedupe bytes saved) * 100
compression-space-saved	Number of bytes saved by compression
compression-space-saved-percent	Percentage of space saved from compression compression bytes saved / (used + compression bytes saved) * 100

To determine the logical space used capacity you would add the values "used + sis-space-saved."

11.3 Compression and Deduplication Options for Existing Data

By default when compression and deduplication are enabled on a volume, all new data will be deduplicated and compressed; however, existing data will not be. Optionally, if you want to deduplicate or compress existing data, you need to run a volume efficiency operation manually, typically only once, shortly after the efficiency is enabled on a volume. This process will run as a low-priority background process scanning through the existing data and compressing first (if compression is selected) and then running deduplication. In general it will create a checkpoint every hour and after each phase of operation. If this operation is stopped you can restart it from the existing checkpoint.

It is possible to select running deduplication only against the existing data. This option will start by reading the existing blocks on disk and creating a fingerprint database. Once the fingerprints are created it will then begin the process of looking for and releasing identical blocks. Refer to the Deduplication Metadata section, above, for more details.

It is also possible to select running compression only against data that existed on disk prior to enabling compression. This option will read and compress existing data while periodically creating checkpoints during its operation. The checkpoint will be deleted in any of the following scenarios.

- When the compression of existing data completes
- Compression is disabled on the volume
- If the `-delete-checkpoint true` option is used while starting or stopping compression of existing data

By default, compression of existing data does not attempt to compress any compression groups that contain shared blocks such as from deduplication or that have blocks locked in Snapshot copies. For volumes that have existing data that contains a lot of deduplicated data or significant amounts of data locked in Snapshot copies, the compression savings from the default use of compression against existing data might be limited. To change the default behavior, the `-shared-blocks true` option can be used to force compression of existing shared blocks created by deduplication or cloning of data.

The `-snapshot-blocks true` option can be used to force compression of existing blocks locked in existing Snapshot copies. Both the `-shared-blocks true` and `-snapshot-blocks true` options can be used in conjunction with each other and each requires that `-scan-old-data true -compression true` be used with it. Running compression on a volume with the `-shared-blocks true` or `-snapshot-blocks true` option can result in increased space usage temporarily.

The reason for the `-snapshot-blocks true` option causing a temporary increase in used disk space is as follows. Compression rewrites the existing data as new compressed blocks. The original data blocks are temporarily not released since they are locked by a Snapshot copy. This temporary increase will be removed when the Snapshot copies that contain the original data blocks are deleted or expire. As an example, if you have a 32k file on disk that is locked in a Snapshot copy, assuming 50% compressibility, you would temporarily have the original 32k on disk plus the newly written compressed version occupying an additional 16k on disk. That results in 48k temporarily on disk. Once the lock is deleted or expires, then the 32k would be freed and only 16k would remain on disk and you would realize your savings. To determine how much temporary space you would require you need to add the size of the Snapshot database and know how compressible the data is.

The reason why the `-shared-blocks true` option may cause a temporary increase in used disk space is as follows. When the volume has shared blocks due to deduplication, compression with the `-shared-blocks true` option will try to compress the shared (deduplicated) blocks. If any of these blocks are compressed, they will be rewritten as new unshared compressed blocks, ostensibly losing any previous savings from deduplication. This temporary increase will remain until deduplication is rerun. If your total compression-only savings are larger than your deduplication-only savings, then this will not be a factor.

11.4 Best Practices for Compressing Existing Data

For optimal savings, some best practices for compressing existing data are the following.

- Remove as many existing Snapshot copies as possible on the volume before compressing existing data. The Snapshot schedule for the volume should be disabled until the compression of existing data completes.
- Compression of existing data can be run with the option `-snapshot-blocks true` to compress data that is locked in Snapshot copies. This option should be used only if the FlexVol volume or data constituents have sufficient space to accommodate the temporary increase required until the locks on the uncompressed blocks from the Snapshot copy are released. Refer to the previous section for more details.
- If existing data in a volume has been deduplicated and you now want to compress this data, including the shared blocks, you should run volume efficiency with the `-scan-old-data true` and `-shared-blocks true` option. This will compress data blocks that are shared due to deduplication followed by redoing deduplication (`volume efficiency start -vserver <SVMname> -volume <volname> -scan-old-data true -shared-blocks true`). This option should be used only if the FlexVol volume or data constituents have sufficient space to accommodate the possible temporary increase of space before deduplication completes.

- Compression of existing data running with the `-shared-blocks true` or `-snapshot-blocks true` on a volume with the autosize option enabled might take up space from the aggregate if the undoing of block sharing utilizes 85% or more of the volume space.
- If using the `-shared-blocks true` or `-snapshot-blocks true` options it is best to run compression on existing data on the source volume of a dump or VSM relationship before the initial baseline is created. If not, after compression of existing data completes with options `-shared-blocks true` or `-snapshot-blocks true` the next update could be significant in size; in essence, it could be like recreating the baseline copy.
- In case the volume has a lot of shared data, such as FlexClone files in VMware environments, NetApp recommends that compression of existing data be run in the default mode, skipping shared blocks to avoid significant FlexVol volume or data constituent inflation.

11.5 Compression and Deduplication Quick Start

This section describes the steps necessary to enable and configure compression and deduplication on a volume.

Table 10) Compression and deduplication quick start.

Action	Command
Enable deduplication on volume	`volume efficiency on -vserver <SVMname> -volume <volname>`
Enable compression on volume; -inline-compression is only required if you want to enable inline-compression	`volume efficiency modify -vserver <SVMname> -volume <volname> - compression true -inline-compression true`
Run initial compression and deduplication scan; not applicable for new volumes	`volume efficiency start -vserver <SVMname> -volume <volname> -scan-old-data true`
Monitor status of the compression and deduplication scan; not applicable for new volumes	`volume efficiency show -vserver <SVMname> -volume <volname> -fields progress`
Assign or change a compression/deduplication policy for a volume	`volume efficiency modify -vserver <SVMname> -volume <volname> -policy <policy_name>`
Manually run compression/deduplication (if not using schedules)	`volume efficiency start -vserver <SVMname> -volume <volname>`

Monitor status of compression and deduplication—not available at the Infinite Volume level, only at the data constituent level within an Infinite Volume.	`FlexVol volume:`
	`volume efficiency show -vserver <SVMname> -volume <volname> -fields progress`
	`Infinite Volume data constituents:`
	`volume efficiency show -vserver <SVMname> -is-constituent true -fields progress`

Monitor compression and deduplication space savings at the FlexVol volume or Infinite Volume level.	`volume show -fields used, sis-space-saved, sis-space-saved-percent, dedupe-space-saved, dedupe-space-saved-percent, compression-space-saved, compression-space-saved-percent`

11.6 Configuring Compression and Deduplication Schedules

By default when a new FlexVol volume or an Infinite Volume is created and deduplication is enabled, the volume is assigned a default compression or deduplication schedule of sun-sat@0. It is possible to set up compression and deduplication to be scheduled to run against new data written to the volume after compression or deduplication has been enabled. It's best to modify the schedule for compression and deduplication to meet your specific needs. This section describes how to configure schedules from the command line (CLI). You can also use System Manager to configure this.

Schedules for compression and deduplication are assigned to a FlexVol volume or an Infinite Volume, not individual data constituents. If you require different schedules for different data constituents, contact NetApp Customer Success Services for assistance. Schedules can be one of the following.

- Time Based: A time-based schedule starts compression/deduplication based on a set timetable.
- Threshold Based: A threshold-based schedule starts compression/deduplication automatically after a set percentage of new data is written to the volume.
- Manual Only: A manual-only schedule will never automatically run compression or deduplication; instead, it needs to be run manually using the `volume efficiency start` command.

For even more control of a time-based schedule you can create a policy that can be used to schedule compression/deduplication. See the section on configuring compression and deduplication schedules using volume efficiency policies, below, for more details.

The following details how to set up each scheduling option for volumes within a cluster.

Assign a Time-Based Schedule to a Volume

A time-based schedule starts compression/deduplication based on a set timetable. This can be useful for volumes that have a known time when there is less activity on the system; compression/deduplication can be scheduled for this time. A time-based schedule can be configured in one of two ways.

```
[day_list][@hour_list]
[hour_list][@day_list]
```

Here is an example of setting vol3's schedule to a time-based schedule set to run on Fridays at midnight.

```
Cluster1::> volume efficiency modify -vserver vsone -volume vol3 –schedule Fri@0

Cluster1::> volume efficiency show -vserver vsone -volume vol3 -fields schedule
  (volume efficiency show)
vserver       volume       schedule
-------       -----------  ------------
vsone         vol3         Fri@0
```

Assign a Threshold-Based Schedule to a Volume

A threshold-based schedule starts compression/deduplication automatically after a specified percentage of new data is written to the volume. This can be useful when your volumes do not have a lot of new data being written to them. It allows you to limit the frequency of running compression/deduplication to when savings are possible. This does not enable you to set compression/deduplication to only happen during scheduled hours such as nonpeak time. You may want to limit this scheduling option to volumes that do not have a lot of new data on systems that do not have heavy peak hours.

Here is an example of setting vol2's schedule to a 20% and 50% threshold value.

```
Cluster1::> volume efficiency modify -vserver vsone -volume vol2 -schedule auto


Cluster1::> volume efficiency show -vserver vsone -volume vol2 -fields schedule
 (volume efficiency show)

vserver       volume       schedule
-------       -----------  -------------
vsone         vol2         auto
```

The default threshold is 20% but this can be changed by adding the @num option to the auto option. Here is an example of how to change the default value from 20% to 50%.

```
Cluster1::> volume efficiency modify -vserver vsone -volume vol2 -schedule
auto@50


Cluster1::> volume efficiency show -vserver vsone -volume vol2 -fields schedule
 (volume efficiency show)

vserver       volume       schedule
-------       -----------  -------------
vsone         vol2         auto@50
```

Assign a Manual-Only Schedule to a Volume

A manual-based schedule is used if you only want compression or deduplication to be run manually. This can be useful if you only want to use inline compression and not postprocess compression or deduplication. The syntax to configure a volume to only run compression/deduplication manually is:

```
Cluster1::> volume efficiency modify -vserver vsone -volume vol1 -schedule -
```

To view the new schedule you can use the volume efficiency show command. Here is an example.

```
Cluster1::> volume efficiency show -vserver vsone -volume vol1 -fields schedule

vserver       volume       schedule
-------       -----------  -------------
vsone         vol1         -
```

11.7 Configuring Compression and Deduplication Schedules Using Volume Efficiency Policies

Another way to assign a time-based schedule for compression and deduplication to a volume using CLI is to use policies. Policies provide more flexibility, more granularity, and simpler management across multiple volumes. Policies are assigned to FlexVol volumes or an Infinite Volume, not individual data constituents. If you require different policies for different data constituents, contact NetApp Customer Success Services for assistance. Once a policy is assigned to a volume, to make changes you simply need to change the policy and the update will happen automatically to all volumes assigned to the policy. Policies also allow the ability to set a duration and starting in clustered Data ONTAP 8.2 the ability to assign volume efficiency priority of either best-effort or background. The ability to set a duration adds the ability to not only assign a start time but also an end time. This can be extremely beneficial for many customers such as:

- Customers who have a significant amount of new data and only a short window to complete compression/deduplication
- Customers who have many scheduled activities that they don't want to have occur in parallel
- Customers with many volumes who all want to run compression/deduplication and want to give each equal time to run during a set window

There are a maximum of three steps to create a policy and assign it to a volume: create a cron job schedule, create an efficiency policy, and assign the efficiency policy to a volume.

The following walks you through the three steps individually. If you already have a cron job schedule or a policy created that meets your needs, skip to step 3.

Step 1: Create a Cron Job Schedule in the Cluster

The first step to set up a policy is to determine that you have a cron job schedule that exists for any time-based policies you want to create. Cron job schedules are created at the cluster level. You can use preexisting cron job schedules or create your own using the command `job schedule cron create`. This job schedule will determine the frequency of and start times for running the job. Here is an example of how to create a job cron schedule that runs every Sunday at 6:05 p.m.

```
Cluster1::> job schedule cron create -name sun605 -dayofweek Sunday -hour 18
-minute 5
```

To view the current cron job schedules:

```
Cluster1::> job schedule cron show
```

Name	Type	Description
5min	cron	@:00,:05,:10,:15,:20,:25,:30,:35,:40,:45,:50,:55
8hour	cron	@2:15,10:15,18:15
daily	cron	@0:10
hourly	cron	@:05
weekly	cron	Sun@0:15
WEmidnight	cron	Sun,Fri,Sat@0:00
sun605	cron	Sun@18:05

Notice that you can create cron job schedules at a more granular level than compression/deduplication schedules. Here you are able to include minutes in the schedule rather than only hours.

Step 2: Create an Efficiency Policy for the Vserver

The next step to set up an efficiency schedule for compression and deduplication is to determine that you have an efficiency policy that meets your needs or to create one. Efficiency policies are created at the Vserver level. You can create, delete, modify, or view efficiency policies with the `volume efficiency policy` command. Here is the usage syntax.

```
Cluster1::> volume efficiency policy ?
  (volume efficiency policy)
  create                     Create  an efficiency policy
  delete                     Delete an efficiency policy
  modify                     Modify an efficiency policy
  show                       Show efficiency policies
```

Here is an example of creating an efficiency policy that will run compression/deduplication every Sunday at 6:05 p.m. for 8 hours using the priority background.

```
Cluster1::> volume efficiency policy create -vserver vsone -policy Sun6pm -
schedule sun605 -enabled true -duration 8 -comment "Run every Sun at 6:05 pm
for 8 hrs" –qos-policy background
```

To see the efficiency policies you have on your cluster you can use the `show` option.

```
Cluster1::> volume efficiency policy show
```

Vserver	Policy Name	Job Schedule	Duration (Hours)	QoS Policy	Enabled	Comment
vsone	Everyday	daily	4	best-effort	true	Run daily at 00:10 am for 4 hours
vsone	Sun6pm	sun605	8	background	true	Run every Sun at 6:05 pm for 8 hrs
vsone	Every8hrs	8hour	1	best-effort	true	Run every 8 hrs starting 2:15 for 1 hr
vsone	Weekends	WEmid	-	background	true	Run Fri at midnight until complete

Notice that again there is more control with a policy in that you can include a stop time/duration.

```
Cluster1::job schedule> delete Sun605pm
```

```
Error: command failed: Cannot delete job schedule. Schedule referenced by 0
      volume Snapshot policy entries, 0 SnapMirror entries, 0 antivirus
      on-demand entries, 1 SIS policy entries, and 0 configuration backup
      settings. Remove references to schedule first.
```

Note: A scheduled job cannot be deleted if it is referenced by a volume efficiency policy. It does not matter if there are no volumes configured with this policy.

Cluster1::job schedule> `cron modify -name defaultEfficiencySched -hour 5`

Warning: Schedule in use by 0 volume Snapshot policy entries, 0 SnapMirror entries, 0 antivirus on-demand entries, 1 SIS policy entries, and 0 configuration backup settings. This change will affect the schedule of all referenced entities. Continue with modify? {y|n}:

Note: Since changes to a policy will take immediate effect on all volumes assigned to the policy, a warning will be given if an attempt is made to modify a job cron schedule referenced by an efficiency policy.

Step 3: Assign an Efficiency Policy to a Volume

The final step to set up an efficiency policy for compression and deduplication is to assign the efficiency policy to volumes.

To add or modify an efficiency policy for a volume, use this syntax.

Cluster1::> `volume efficiency modify -vserver vsone -volume vol4 -policy Sun605pm`

Cluster1::> `volume efficiency modify -vserver vsone -volume vol5 -policy Sun605pm`

You can assign multiple volumes within the same storage virtual machine to use the same policy.

Here is the syntax to see the current efficiency policy or schedule that is assigned to each volume.

Cluster1::> `volume efficiency show -fields policy, schedule`

vserver	volume	policy	schedule
vsone	vol1	-	-
vsone	vol2	-	auto@50
vsone	vol3	-	Fri@0
vsone	vol4	Sun605pm	-
vsone	vol5	Sun605pm	-

In summary you can have either a schedule or a policy assigned to a volume. You can still run compression/deduplication manually regardless of what schedule or policy is assigned. Policies provide the following extra benefits over a compression/deduplication schedule.

- Minutes: Add the ability to schedule at a particular minute, not just on the hour.
- Stop Times: Add the ability to include duration so you can schedule not only a start time but also a stop time. If compression/deduplication doesn't complete in the specified duration it will stop, create a checkpoint, and continue at the next scheduled time.
- Volume efficiency priority: Starting in clustered Data ONTAP 8.2 you have the ability to assign a priority of either best-effort or background to a volume efficiency operation.
- Simpler Management: Allows one simple change to a policy to change multiple volumes rather than having to modify each volume individually.

Note: NetApp recommends setting volumes with a volume efficiency priority of background be scheduled to run at different times than volumes with a volume efficiency priority of best-effort. Typically volumes associated with the best-effort priority require volume efficiency operations to run at specific times to make sure maximum effort is put into achieving savings. Volumes associated with background priority are typically those where performance is of utmost importance and savings can be opportunistic, and so savings will be attempted when there are available resources, but there is no guarantee about how quickly this will occur. Given this and given that you can only have a maximum of eight volume efficiency operations running in parallel, it is important to schedule these two types of volume efficiency operations to run at separate times. This will help avoid the situation where volumes associated with best-effort priority are queued for long periods of time while the volumes associated with background priority continue to run.

11.8 End-to-End Compression and Deduplication Examples

This section describes the steps necessary to enable and configure compression and deduplication on a volume using clustered Data ONTAP 8.1 or later.

The first example describes the process of creating a new FlexVol volume and then configuring, running, and monitoring compression and deduplication on it.

The second example describes the process of adding compression and deduplication savings to an already existing FlexVol volume that contains data. We then run compression and deduplication to get savings on the already existing data on disk.

Note: The steps are spelled out in detail, so the process appears much longer than it would be in the real world.

Example One: Creating a New Volume and Enabling Inline Compression and Deduplication

This example creates a new volume named volArchive in a storage virtual machine called vsone.

1. Create a FlexVol volume (no larger than the maximum volume size limit for your system).

Cluster1::> `vol create -vserver vsone -volume volArchive -aggregate aggr1 -size 200GB`
   (volume create)
[Job 1545] Job succeeded: Successful

2. Enable deduplication on the FlexVol volume (`volume efficiency on`), followed by compression (volume efficiency modify –compression true –inline-compression true) (-inline-compression is only required if you want to use inline compression), and verify that it is turned on. The `volume efficiency show` command shows the compression and deduplication configuration for FlexVol volumes.

   After you turn deduplication on, Data ONTAP lets you know that if this were an existing FlexVol volume that already contained data before deduplication was enabled, you could run `volume efficiency start –scan-old-data`. In this example it's a brand-new FlexVol volume, so that's not necessary.

Cluster1::> `volume efficiency on –vserver vsone –volume volArchive`
   (volume efficiency on)
Efficiency for volume 'volArchive' of Vserver 'vsone' is enabled.
Already existing data could be processed by running "volume efficiency start -vserver vsone -volume volArchive -scan-old-data true".

Cluster1::> `volume efficiency modify -vserver vsone -volume volArchive -compression true -inline-compression true`
 (volume efficiency modify)

Cluster1::> `volume efficiency show -vserver vsone -volume volArchive`

    Vserver Name: vsone
    Volume Name: volArchive
    Volume Path: /vol/volArchive
    State: Enabled
    Status: Idle
    Progress: Idle for 00:05:00
    Type: Regular
    Schedule: sun-sat@0
    Efficiency Policy Name: -
    Min Blocks Shared: 1
    Blocks Skipped Sharing: 0
    Last Operation State: Success
    Last Success Operation Begin: Mon Jul 25 22:31:22 GMT 2011
    Last Success Operation End: Mon Jul 25 22:31:22 GMT 2011
    Last Operation Begin: Mon Jul 25 22:31:22 GMT 2011
    Last Operation End: Mon Jul 25 22:31:22 GMT 2011
    Last Operation Size: 0B
    Last Operation Error: -
    Change Log Usage: 0%
    Logical Data Size: 196KB
    Logical Data Limit: 50.00TB
    Logical Data Percent: 0%
    Queued Job: -
    Stale Fingerprint Percentage: 0
    Compression: true
    Inline Compression: true

3. Turn off the deduplication schedule.

Cluster1::> `volume efficiency show -vserver vsone -volume volArchive -fields schedule`
vserver   volume   schedule
-------   ----------   ---------
vsone   volArchive   sun-sat@0

Cluster1::> `volume efficiency modify -schedule - -vserver vsone -volume volArchive`

Cluster1::> `volume efficiency show -vserver vsone -volume volArchive -fields schedule`
vserver   volume   schedule
-------   ----------   --------
vsone   volArchive   -

4. Mount the FlexVol volume and copy data into the new archive directory FlexVol volume.

5. Examine the FlexVol volume. Use the `df -S` command to examine the storage consumed and the space saved. Note that only compression savings have been achieved so far by simply copying data to the FlexVol volume, even though deduplication is also turned on. What has happened is that the inline compression compressed the new data as it was written to the volume. Since deduplication was enabled, all the new blocks have had their fingerprints written to the change log file. Until deduplication is actually run, the duplicate blocks will not be removed.

```
Cluster1::> df -S volArchive
```

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed	%compressed	Vserver
/vol/volArchive/	139178264	36126316	21%	0	0%	36126316	21%	vsone

6. Manually run postprocess compression and deduplication on the FlexVol volume. This causes the compression engine to compress any blocks that were skipped by inline compression followed by processing of the change log, which includes fingerprints to be sorted and merged and duplicate blocks to be found.

```
Cluster1::> volume efficiency start -vserver vsone -volume volArchive
```

 (volume efficiency start)

The efficiency operation for volume 'volArchive' of Vserver 'vsone' has started.

7. Use `volume efficiency show` to monitor the progress of compression and deduplication operations.

```
Cluster1::> volume efficiency show -vserver vsone -volume volArchive -fields progress
```

(volume efficiency show)

vserver	volume	progress
vsone	volArchive	"23 GB (77%) Compressed"

```
Cluster1::> volume efficiency show -vserver vsone -volume volArchive -fields progress
```

(volume efficiency show)

vserver	volume	progress
vsone	volArchive	"164 GB Searched"

```
Cluster1::> volume efficiency show -vserver vsone -volume volArchive -fields progress
```

(volume efficiency show)

vserver	volume	progress
vsone	volArchive	"39 GB (43%) Done"

Cluster1::> `volume efficiency show -vserver vsone -volume volArchive -fields progress`

(volume efficiency show)

vserver	volume	progress
vsone	volArchive	"Idle for 00:02:20"

8. When `volume efficiency show` indicates that the FlexVol volume is once again in the Idle state, compression and deduplication have finished running, and you can check the space savings they provided in the FlexVol volume.

Cluster1::> `df -S`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed	%compressed	Vserver
/vol/volArchive/	70313	103348640	59%	64598736	47%	38749904	35%	vsone

9. Adjust the compression and deduplication schedule or policy as required in your environment. Refer to the section on [configuring compression and deduplication schedules](#) for more specifics.

That is all there is to it.

Example Two: Enabling Compression and Deduplication on an Existing Volume

This example adds compression and deduplication savings to an existing FlexVol volume with data already on the volume. Although it is not necessary, this example includes the steps involved if you want to compress and deduplicate the data already on disk, in addition to the new writes to disk.

1. Enable deduplication on the FlexVol volume, followed by compression (-inline-compression is optional), and verify that it is turned on. The volume efficiency show command shows the compression and deduplication configuration for FlexVol volumes.

Cluster1::> `volume efficiency on -vserver vsone -volume volExisting`
  (volume efficiency on)
Efficiency for volume 'volExisting' of Vserver 'vsone' is enabled.
Already existing data could be processed by running "volume efficiency start -vserver vsone -volume volExisting -scan-old-data true".

Cluster1::> `volume efficiency modify -vserver vsone -volume volExisting -compression true -inline-compression true`
   (volume efficiency modify)

```
Cluster1::> volume efficiency show -vserver vsone -volume volExisting

                   Vserver Name: vsone
                    Volume Name: volExisting
                    Volume Path: /vol/volExisting
                          State: Enabled
                         Status: Idle
                       Progress: Idle for 00:01:06
                           Type: Regular
                       Schedule: sun-sat@0
         Efficiency Policy Name: -
              Min Blocks Shared: 1
         Blocks Skipped Sharing: 0
           Last Operation State: Success
 Last Success Operation Begin: Mon Jul 25 22:57:27 GMT 2011
   Last Success Operation End: Mon Jul 25 22:57:27 GMT 2011
           Last Operation Begin: Mon Jul 25 22:57:27 GMT 2011
             Last Operation End: Mon Jul 25 22:57:27 GMT 2011
            Last Operation Size: 0B
           Last Operation Error: -
               Change Log Usage: 0%
              Logical Data Size: 196KB
             Logical Data Limit: 50.00TB
           Logical Data Percent: 0%
                     Queued Job: -
    Stale Fingerprint Percentage: 0
                    Compression: true
             Inline Compression: true
```

2. Examine the FlexVol volume. Use the `df -S` command to examine the storage consumed and the space saved.

```
Cluster1::> df -S volExisting
File system        used      total-saved  %total-saved  deduplicated  %deduplicated  compressed
%compressed   Vserver
/vol/volExisting/  173952092    0            0%            0              0%             0
0%            vsone
```

Decision: Compress existing data or not?
At this time only new data will be compressed and have fingerprints created.
No: If you want to only compress/deduplicate new writes, you can skip to step 9.
Yes: If you want to compress/deduplicate existing data on disk, the following additional steps are required.

3. Disable the postprocess compression and deduplication schedule.

```
Cluster1::> volume efficiency show -vserver vsone -volume volExisting -fields
schedule
vserver          volume          schedule
----------       ----------      ------------
vsone            volExisting     sun-sat@0


Cluster1::> sis modify -schedule - -vserver vsone -volume volExisting
```

```
Cluster1::> sis show -vserver vsone -volume volExisting -fields schedule
vserver         volume          schedule
---------       ---------       ------------
vsone           volExisting     -
```

4. Record the current Snapshot schedule for the volume. Change the Snapshot policy so that Snapshot copies are not scheduled to be taken.

```
Cluster1::> volume show -vserver vsone -volume volExisting -fields snapshot-
policy
vserver         volume          snapshot-policy
---------       ---------       --------------------
vsone           volExisting     default
```

```
Cluster1::> volume modify -vserver vsone -volume volExisting -snapshot-policy
none
```
Warning: You are changing the Snapshot policy on volume volExisting to none. Any Snapshot copies on this volume from the previous policy will not be deleted by this new Snapshot policy.
Do you want to continue? {y|n}: y

5. Delete as many Snapshot copies as possible.

```
Cluster1::volume> snap show -vserver vsone -volume volExisting
                                                                ---Blocks---
Vserver  Volume  Snapshot                               Size Total% Used%
-------- ------- -------------------------------- ------------ ------ -----
vsone    volExisting
                 hourly.2011-07-25_2305                 80KB    0%    27%
                 snap.1                                 92KB    0%    30%
                 snap.2                                 80KB    0%    27%
3 entries were displayed.
```

```
Cluster1::volume> snap delete -vserver vsone -volume volExisting -snapshot snap.1
```

```
Cluster1::volume> snap delete -vserver vsone -volume volExisting -snapshot snap.2
```

```
Cluster1::volume> snap delete -vserver vsone -volume volExisting -snapshot
hourly.2011-07-25_2305
```

```
Cluster1::volume> snap show -vserver vsone -volume volExisting
```
There are no entries matching your query.

6. Start compression and deduplication of the existing data on the volume (run this during low system usage times). You can run with the option `-dedupe true` if you only want to run deduplication on the existing data. You can run with the `-compression true` option if you only want to compress the existing data; however, the next time the deduplication process runs it will deduplicate the existing data.

```
Cluster1::volume> volume efficiency start -vserver vsone -volume volExisting -
scan-old-data true
```
Warning: The file system will be scanned to process existing data in volume 'volExisting' of Vserver 'vsone'. This operation may initialize related existing metafiles. Do you want to continue? {y|n}: y
The efficiency operation for volume 'volExisting' of Vserver 'vsone' has started.

7. Use `volume efficiency show` to monitor the progress of compression and deduplication.

Cluster1::> `volume efficiency show -vserver vsone -volume volExisting -fields progress` (volume efficiency show)

vserver	volume	progress
vsone	volExisting	"122 GB Scanned, 25 GB Compressed "

Cluster1::> `volume efficiency show -vserver vsone -volume volExisting -fields progress` (volume efficiency show)

vserver	volume	progress
vsone	volExisting	"164 GB Searched"

Cluster1::> `volume efficiency show -vserver vsone -volume volExisting -fields progress` (volume efficiency show)

vserver	volume	progress
vsone	volExisting	"41 GB (45%) Done "

Cluster1::> `volume efficiency show -vserver vsone -volume volExisting -fields progress` (volume efficiency show)

vserver	volume	progress
vsone	volExisting	"Idle for 00:02:43"

8. When `volume efficiency show` indicates that the FlexVol volume is once again in the Idle state, compression and deduplication have finished running, and you can check the additional space savings they provided in the FlexVol volume.

Cluster1::> `df -S`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed
%compressed	Vserver					
/vol/volExisting/	72005148	103364200	59%	64621288	47%	38742912
35%	vsone					

9. Reconfigure the Snapshot policy for the volume.

Cluster1::> `volume modify -vserver vsone -volume volExisting -snapshot-policy default`
Warning: You are changing the Snapshot policy on volume volExisting to default. Any Snapshot copies on this volume from the previous policy will not be deleted by this new Snapshot policy.
Do you want to continue? {y|n}: `y`

10. Adjust the compression/deduplication schedule or policy as required in your environment. Refer to the section on configuring compression and deduplication schedules for more specifics.

That's all there is to it.

11.9 Volume Efficiency Priority

If all of the nodes in the cluster are running a minimum of clustered Data ONTAP 8.2, storage QoS provides administrators the choice of running a volume efficiency operation with a priority of either "best effort" or "background". The priority defines whether the volume efficiency operation will or won't have equal footing with user workloads and other system processes (not running in background). You can set volume efficiency priority on a volume-by-volume basis.

By default, volume efficiency operations run with a priority of "best-effort". This allows volume efficiency operations to compete for resources on an equal footing with user workloads and other system processes not running in background. This is the same as it was in Data ONTAP releases prior to 8.2. Using this priority will allow deduplication and compression to complete the most quickly; however, there might be some impact on the performance of client I/O on the system. The best-effort volume efficiency priority is recommended for volumes where achieving maximum savings is the most important. Volume efficiency policies using a priority of best-effort should be scheduled to run during nonpeak hours to reduce the possibility of impact on other applications.

Running a volume efficiency operation with the priority "background" will allow this process to use leftover resources not used by user workloads and other system processes not running in background; as a result, this might take longer to complete but should have less effect on client I/O. The volume efficiency priority of background maximizes system resource usage and is recommended for volumes where performance is of utmost importance.

A total of eight volume efficiency operations can run in parallel, whether with a priority of "background" or "best effort". NetApp recommends setting volumes with a volume efficiency priority of background to run at different times than volumes with a volume efficiency priority of best-effort. Typically volumes associated with the volume efficiency priority of best-effort require volume efficiency operations to run at specific times to make sure maximum effort is put into achieving savings. Volumes associated with a volume efficiency priority of background are typically those where performance is of utmost importance and savings can be opportunistic, and so savings will be attempted when there are available resources but there is no guarantee about how quickly this will occur. Given this, it is important to schedule these two types of volume efficiency operations to run at separate times to avoid the situation where volumes associated with best-effort priority are queued for long periods of time while the volumes associated with background priority continue to run.

Volume efficiency priority can be assigned to volume efficiency operations in either the volume efficiency start command or in the volume efficiency policy. It is not recommended to manually run volume efficiency operations with a priority of background. This is especially important on systems that are very busy since the volume efficiency operations running in the background can take a very long time to complete and will limit the number of volume efficiency operations set with a priority of best-effort that can run in parallel.

12 Transition

This section talks about transitioning from a system running Data ONTAP operating in 7-Mode to a system running clustered Data ONTAP. There are two main ways to migrate the data: doing a copy-based migration or a replication-based migration. A copy-based migration is a logical transfer of the data using a basic file copy method. A replication-based migration is a physical transfer using NetApp volume SnapMirror technology. There are a few changes and improvements to be aware of if the volumes on the 7-Mode system had deduplication or compression enabled on them.

- Deduplication and compression move from sis command structure to volume efficiency set of commands. The 7-Mode commands will still work in clustered Data ONTAP but do not offer as many options. Familiarize yourself with the new commands that are available in clustered Data ONTAP. You can see these in the command summary section of this document.
- If you have any sis scripts that you want to use on the new volume, they will continue to work, but you should update them to the new scripts at a convenient time to make use of the more robust clustered Data ONTAP sets of commands.
- Consider making use of volume efficiency policies instead of deduplication and compression schedules to make use of the duration and increased flexibility in start times.

12.1 Copy-Based Migration

There is nothing specifically required to plan for if you are performing a copy-based migration of data from a deduplication or compression volume running on Data ONTAP operating in 7-Mode to clustered Data ONTAP. How the copy-based migration works is by performing a logical copy of a file through a host. You are able to migrate data regardless of whether the source had deduplication or compression.

Some things of which to be aware when performing a copy-based migration from a volume running on Data ONTAP operating in 7-Mode if the source had either compression or deduplication enabled include:

- Compression/deduplication space savings are always lost during the network copy; therefore, there are no network bandwidth savings.
- Compression/deduplication will have to be redone on the destination in order to regain space savings.
- Both source and destination storage systems run compression and deduplication independently of each other.
- The compression/deduplication schedule is not tied to the migration. In other words, the compression/deduplication process does not automatically start at the completion of the migration.

Since savings are not preserved on the destination and it is possible to have more logical data than physical space on disk, it is important to understand the transfer size and make sure that your destination is large enough to support the update. Let's look at an example.

Source volume → 100TB in size

150TB of logical data

75TB of physical data

- 20% savings from compression (30TB compression savings)
- 30% savings from deduplication (45TB deduplication savings)

Destination volume → 100TB in size

This copy-based migration would fail because the volume will need to be able to write 150TB of data to the volume before volume efficiency operations can run to regain the savings. If you had inline compression enabled, you would still need to be able to write 120TB of data to the volume before running volume efficiency operations to regain the additional deduplication savings.

As a best practice NetApp recommends stopping any active sis operations as well as disabling the sis schedule on the source volume before initiating a copy-based migration.

Remember, same as you would when creating a new volume, you will want to set up a schedule or policy for when deduplication and compression run on the destination volume.

12.2 Replication-Based Migration

There is nothing specific required if you are performing a replication-based migration of data from a deduplication or compression volume running on Data ONTAP operating in 7-Mode to clustered Data ONTAP. How the replication-based migration works is it performs a volume SnapMirror copy from the 7-Mode source volume to a clustered Data ONTAP destination FlexVol volume. You are able to migrate data regardless of whether the source had deduplication or compression. The replication-based migration based on SnapMirror operates at a physical block level; thus, when there are compression and deduplication savings on the source, the data sent over the wire for replication remains compressed and deduplicated, and therefore the savings are inherited at the destination. This can significantly reduce the time for the transfer to complete as well as the amount of network bandwidth required during the replication-based migration. The following combinations of compression/deduplication are supported during replication-based migration.

Table 11) Supported compression and deduplication configurations for replication-based migration.

Storage Efficiency Enabled on the Source Volume	Storage Efficiency Enabled on the Destination Volume		
	Deduplication	Deduplication + Compression	Neither
Deduplication	✓	✗	✗
Deduplication + compression	✗	✓	✗
Neither	✗	✗	✓

To successfully perform a replication-based migration of data within a volume with deduplication and/or compression savings:

- Both source and destination systems should use an identical release of Data ONTAP. If this is not possible, then the destination must be a newer version than the source.

- If you use compression on both the source and the destination volumes, they must both be contained within a 64-bit aggregate.

- Both compression and deduplication can only be managed on the source system—the FlexVol volume at the destination system inherits all the compression/deduplication storage savings.

- As a best practice, start the replication-based migration of a compressed/deduplicated volume after compression/deduplication is complete (that is, not in the middle of the deduplication or compression process). This is to avoid sending uncompressed/undeduplicated data and additional temporary metadata files over the network.

- Do not start compression or deduplication of existing data on source while the replication-based migration is occurring.

- Check to see if the deduplication schedule meets your needs on your cluster and consider switching to a volume efficiency policy instead.

Deduplication maintains a partially ordered fingerprint database in the FlexVol volume along with the aggregate copy. Because of this, the destination system will not only have the space savings from the source volume but will also have a copy of the ordered fingerprint database. After a replication-based migration completes and you break the mirror, the first time volume efficiency runs on the new volume, the aggregate fingerprint database will be automatically constructed from the copy in the FlexVol volume on the destination volume. This can result in a one-time increase in the time for volume efficiency operations to complete.

13 Upgrading and Reverting

Major and minor nondisruptive upgrades to clustered Data ONTAP are supported with both compression and deduplication. See the feature interoperability section on [Nondisruptive upgrades](#) for details.

13.1 Upgrading to a Newer Version of Clustered Data ONTAP

Upgrading to a newer version of clustered Data ONTAP is a nondisruptive operation regardless of whether the volumes have deduplication enabled or not. However, you should be aware of the fact that part of an upgrade to a newer major release on a deduplication-enabled volume or data constituent includes upgrading the deduplication metadata, and as a result you might notice that volume efficiency operations take longer to complete during this operation. Note that this is a one-time increase on each FlexVol volume or data constituent that had deduplication enabled prior to the upgrade and will only happen after an upgrade; subsequent volume efficiency operations will complete as expected. To understand the impact of the volume efficiency operations, see the [performance](#) section for more details. The process of upgrading the deduplication metadata is done in two stages.

The first stage involves creating new change log metafiles in the FlexVol volume or data constituents and is done automatically during the upgrade itself, at boot time, while the volume is coming online.

The second stage involves sorting the change logs and fingerprints created from the previous version and recreating with the new format. This is done automatically the first time volume efficiency operations are run on each FlexVol volume or data constituent that had deduplication enabled prior to the upgrade. The second step can take a significant amount of time to complete, during which you can see an increased amount of CPU on your system and see the progress of the upgrade using the `volume efficiency show –instance` command. The amount of time for this to complete might be different for each FlexVol volume and data constituent and mainly depends on the size of the change log and fingerprint database, the storage efficiency priority being used, and the amount of available resources. During this phase there will not be any checkpoints so if you need to stop deduplication you should wait for this upgrade process to complete first.

13.2 Reverting to an Earlier Version of Clustered Data ONTAP

When you're considering reverting a system to an earlier version of clustered Data ONTAP, we typically recommend contacting NetApp Customer Success Services for assistance.

Since neither compression nor deduplication is supported in versions earlier than clustered Data ONTAP 8.1, you have to undo all compression and deduplication savings prior to reverting to an earlier release.

When reverting a system from clustered Data ONTAP 8.2 to clustered Data ONTAP 8.1.x, it will require converting the deduplication metadata to an earlier format as well as reducing the amount of logical data in the volume to within the clustered Data ONTAP 8.1 limits. The limit for logical data in clustered Data ONTAP 8.1 is equal to the maximum volume size supported on your platform; see the section on Maximum Logical Data Size Processing Limits for more details.

Scenarios in which customers can revert their NetApp storage systems without guidance from NetApp Customer Success Services include:

- If the NetApp storage system is new and has not yet been used, then the `volume efficiency revert_to` command can be used without guidance from NetApp Customer Success Services.

- If the NetApp storage system is being used as a test system and there is no production data present, then the `volume efficiency revert_to` command can be used without guidance from NetApp Customer Success Services.

Reverting in any other scenario should not be attempted without assistance from NetApp Customer Success Services: http://www.netapp.com/us/support/ngs-contacts.html.

The `volume efficiency revert_to` command will provide sufficient guidance to successfully complete the revert process, including the process for reverting deduplicated volumes, in these scenarios. If you encounter problems during or after the revert process, contact NetApp Customer Success Services.

To check the progress of revert_to operations, use the command `volume efficiency show -instance`. For example:

fas6070-ppe02> `volume efficiency show -instance -vserver vs1 -volume vol1`

Vserver Name: vs1

Volume Name: vol1

Volume Path: /vol/vol1

State: Enabled

Status: Downgrading

Progress: Building metadata and expanding dedupe data: 1TB Processed

The `volume efficiency revert_to` operation will perform first will determine if it is necessary to expand some deduplicated data to reduce the amount of logical data that was supported in the previous version to which you are downgrading and if so perform this expansion. Next, if deduplication is enabled and compression is disabled, it will downgrade the fingerprint database and change log files using a series of sort and merge operations. If compression is enabled, revert_to will delete all of the deduplication metadata and generate empty change logs and then will rebuild it from scratch.

14 Compression and Deduplication with Other NetApp Features

NetApp has several products that work with both compression and deduplication. This section discusses how compression and deduplication interact with other NetApp products and features. It gives details regarding support, limitations, and best practices.

14.1 Data Protection

Snapshot Copies

Snapshot copies lock blocks on disk that cannot be freed until the Snapshot copy expires or is deleted. Once a Snapshot copy of data is made on any volume, any subsequent changes to data contained in that Snapshot copy will temporarily require additional disk space until the Snapshot copy is deleted or expires. The same is true with deduplication- and/or compression-enabled volumes.

Space savings will not be seen on deduplicated volumes initially if the blocks are locked by Snapshot copies. This is because the blocks are not freed until the lock is removed. Postprocess compression will rewrite the uncompressed blocks as new blocks on disk. If these original blocks are locked in a Snapshot copy, then both the original and newly compressed blocks will temporarily exist on disk until the Snapshot copy expires or is deleted. With inline compression, the only way that inline compression space savings are affected by Snapshot copies is for overwrites.

Some best practices to achieve the best space savings from deduplication- and/or compression-enabled volumes that contain Snapshot copies are:

- Run compression and deduplication before creating new Snapshot copies.
- Wait for postprocess compression and deduplication to complete before creating new Snapshot copies.
- If running deduplication without compression, schedule deduplication only after significant new data has been written to the volume.
- Limit the number of Snapshot copies you maintain.
- If possible, reduce the retention duration of Snapshot copies.
- Configure appropriate reserve space for the Snapshot copies.

NetApp's compression solution also includes the ability to compress data that existed on disk prior to enabling compression on the volume. By default, existing data locked within a Snapshot copy is skipped.

Specific best practices for compression of existing data include:

- Delete as many Snapshot copies as possible before running compression against existing data.
- Change the Snapshot policy to none until compression of existing data completes.

Volume SnapMirror

Volume SnapMirror allows you to replicate data to another location for disaster recovery purposes. Both compression and deduplication are supported with volume SnapMirror. Volume SnapMirror operates at the physical block level; thus, when compression and deduplication are enabled on the source, the data sent over the wire for replication remains compressed/deduplicated and therefore the savings are inherited at the destination. This means that shared blocks are only transfer once, compression savings are retained typically resulting in smaller transfer sizes. This can significantly reduce the amount of network bandwidth required as well as the time for the replication to complete. The following combinations of volume SnapMirror and compression/deduplication are supported.

Table 12) Supported compression and deduplication configurations for volume SnapMirror.

Storage Efficiency Enabled on the Source Volume	Storage Efficiency Enabled on the Destination Volume		
	Deduplication	Deduplication + Compression	Neither
Deduplication	✓	✗	✗
Deduplication + Compression	✗	✓	✗
Neither	✗	✗	✓

To run deduplication and/or compression with volume SnapMirror:

- Both source and destination systems should use an identical release of Data ONTAP. If this is not possible, then the destination must be a newer version than the source.

- If you use compression on both the source and the destination volumes they must both be contained within a 64-bit aggregate.

- Both compression and deduplication can only be managed on the source system—the FlexVol volume at the destination system inherits all the compression/deduplication attributes and storage savings.

- The volume SnapMirror update schedule is not tied to the compression/deduplication schedule.

- When configuring volume SnapMirror and compression/deduplication, it is important to consider the compression/deduplication schedule and the time you want to start a volume SnapMirror initialization. As a best practice, start volume SnapMirror initialization of a compressed/deduplicated volume after compression/deduplication is complete (that is, not in the middle of the deduplication or compression process). This is to avoid sending uncompressed/undeduplicated data and additional temporary metadata files over the network. If the temporary metadata files in the source volume are locked in Snapshot copies, they also consume extra space in the source and destination volumes.

- It is especially important when preparing to initialize a volume SnapMirror relationship on a compressed or deduplicated volume to decide if you want to perform the optional operation of compressing or deduplicating existing data on the primary, especially if using the `-shared-blocks true` or `-snapshot-blocks true` options. This is because running compression against existing data with the `-shared-blocks true` or `-snapshot-blocks true` options can result in a large number of physical-level changes on disk. If you are deduplicating existing data, this can result in a large number of logical changes on disk. For both compression and deduplication of existing data, this means that SnapMirror recognizes these as changed blocks and includes them in their next data transfer to the destination volume. As a result, volume SnapMirror updates that occur after deduplication or compression of existing data with the `-shared-blocks true` or `-snapshot-blocks true` options are likely to be much larger than normal. For preexisting volume SnapMirror relationships, it is important to take into consideration the big surge of data involved in the transfer after compression or deduplication of existing data and to plan accordingly. See the section on [compression and deduplication options for existing data](#), earlier, for more details.

In case of a disaster at the primary location, you might need to break the volume SnapMirror relationship and have the volume SnapMirror destination start serving data. Deduplication maintains a partially ordered fingerprint database in the FlexVol volume and data constituents along with the aggregate copy. Because of this the destination system will not only have the space savings from the source volume but will also have a copy of the ordered fingerprint database. After a volume SnapMirror break the aggregate fingerprint database will be automatically constructed from the copy in the FlexVol volume or data constituents during the first deduplication operation on the destination volume.

SnapVault

Both compression and deduplication are supported with SnapVault. The behavior of compression and deduplication with SnapVault is dependent upon the efficiency settings on the source and destination FlexVol volumes. If compression is or ever has been enabled on the SnapVault destination, then the savings from the source are not retained over the network transfer, but they can be regained. All other configurations will retain the savings over the network and on the destination. Table 13 is a quick summary of the behavior.

Table 13) When deduplication and compression savings are retained on SnapVault destinations.

Source FlexVol Volume	SnapVault Destination	Savings Retained over the Network and on Destination	Additional Activities?
Neither deduplication nor compression is enabled	Neither deduplication nor compression is enabled	Yes	No
Neither deduplication nor compression is enabled	Deduplication enabled	Yes	Yes, deduplication automatically runs on destination after transfer is complete to gain deduplication savings
Neither deduplication nor compression is enabled	Compression + deduplication enabled	No	Yes, compression and deduplication automatically run on destination after transfer is complete to gain savings
Deduplication enabled	None	Yes	No
Deduplication enabled	Deduplication enabled	Yes	Yes, deduplication automatically runs on destination after transfer is complete to try to gain additional deduplication savings
Deduplication enabled	Compression + deduplication enabled	No	Yes, compression and deduplication automatically run on destination after transfer is complete to gain savings
Compression + deduplication enabled	None	Yes	No
Compression + deduplication enabled	Deduplication enabled	Yes	Yes, deduplication automatically runs on destination after transfer is complete to try to gain additional deduplication savings
Compression + deduplication enabled	Compression + deduplication enabled	No	Yes, compression and deduplication automatically run on destination after transfer is complete to gain savings

As a best practice, if deduplication is enabled on the source, only enable deduplication on the SnapVault destination if you expect SnapVault updates to occur from Snapshot copies created before deduplication has completed on the source.

As a best practice, only enable compression on the SnapVault destination if you can't run compression on the source.

If you have compression or deduplication enabled on the destination, it is started automatically after the transfer is complete. You can't change when this runs; however, you can change the volume efficiency priority that is assigned to the volume.

Here are some recommendations for SnapVault destinations when the source has compression enabled:

- If you require compression savings on the destination and your source also has compression enabled, then do not enable compression on the SnapVault destination. The savings will already be inherited on the destination. Second, if you enable compression on the SnapVault destination, then the savings will be lost during the transfer, and you will have to redo savings on the destination. Further, if you ever enable it on the destination, even if you later disable it, you will never retain the savings from the source.

- Postprocess compression of existing data results in physical-level changes to data. This means that SnapVault recognizes these as changed blocks and will include them in their data transfers to the destination volume. As a result, SnapVault transfers that follow a `volume efficiency start –scan-old-data true` command are likely to be much larger than normal. If possible, NetApp recommends compressing existing data on the source prior to running baseline transfers for SnapVault. For preexisting SnapVault relationships, it is important to take into consideration the big surge of data involved in the transfer and to plan accordingly.

Here are some recommendations for SnapVault destinations when the source doesn't have deduplication or compression enabled but the destination does:

- If you require compression savings on the destination and your source doesn't have compression enabled, then NetApp recommends using inline compression if you have sufficient CPU resources during the backup. Inline compression will provide the maximum savings without having any temporary impact on the Snapshot copy size. If you use postprocess compression without inline compression, then the size of the archival Snapshot copy will temporarily be larger because it contains the uncompressed version of the blocks just transferred. This temporary increase in Snapshot space will be removed after the deduplication process completes and a newer archival Snapshot copy is created. Also, be aware that once you enable compression on the destination, even if you later disable it and enable it on the source, you will never retain the savings from the source.

- After a SnapVault transfer is complete, it automatically creates an archival Snapshot copy. If you are running deduplication or compression on the destination, the archival Snapshot copy is replaced with a new one after compression/deduplication finishes running on the destination. (The name of this new Snapshot copy is the same as that of the archival copy, but the creation time of this copy is changed.)

- The compression/deduplication schedule on the destination is tied to the SnapVault schedule and cannot be modified. The compression/deduplication schedule on the source is not tied to the SnapVault update schedule, and it can be configured independently on a volume.

- By default, every SnapVault update (baseline or incremental) kicks off the compression/deduplication process on the destination after the archival Snapshot copy is created if compression/deduplication is enabled. This can be modified by setting the postprocess compression/deduplication schedule to manual. By configuring the postprocess compression and deduplication schedule to manual, it prevents deduplication metadata from being created and stops the postprocess compression and deduplication processes from running. This option is beneficial for SnapVault destinations that have data with good compression savings, but minimal deduplication savings.

- For maximum savings the idd (incompressible data detection) option should be set to false on a SnapVault destination if the volume efficiency schedule is set to manual.

- The compression/deduplication schedule on a SnapVault destination cannot be modified to run at a specific time or after a specific amount of data has been written to the volume. The `volume efficiency start` command is not allowed; however, the `volume efficiency start –scan-old-data true` command can be run manually on the destination.

- The SnapVault update is not tied to the compression/deduplication operation; that is, a subsequent incremental update is allowed to run even if the compression/deduplication process on the destination volume from the previous backup is not complete. In this case, the compression/deduplication process will create a checkpoint and stop running during the SnapVault update and resume after the SnapVault update is complete however, the archival Snapshot copy will not be replaced. This can result in the active file system temporarily having less savings and the Snapshot copy taking more space until the Snapshot copy expires.

You can manage deduplication and compression on SnapVault destinations using either the CLI or OnCommand products. For additional SnapVault information, refer to TR-3487: SnapVault Design and Implementation Guide.

SnapRestore

SnapRestore® functionality is supported with both compression and deduplication, and it works in the same way with either feature as it does without. When you initiate a SnapRestore operation on a volume, the restored data retains the original space savings. Deduplication metadata will also be restored to metadata corresponding to the data maintained in the Snapshot copy. This will require some additional processing during the first deduplication operation completed after using volume SnapRestore.

14.2 Other NetApp Features

Infinite Volume

Both compression and deduplication are supported on Infinite Volumes. Compression and deduplication operate the same way on Infinite Volumes as they do on FlexVol volumes. Here are some important things to consider.

- Compression and deduplication are enabled at the Infinite Volume level. This means that if you want to enable either deduplication or data compression it is enabled on the entire Infinite Volume; it can't be enabled on some data constituents and not others.
- When deduplication is run it will look for duplicate blocks within a single data constituent, not across the entire Infinite Volume. Blocks that exist in different data constituents are not deduplicated against one another. As an example, if you have four data constituents, each with four copies of the same block, after deduplication completes on the Infinite Volume each data constituent will store one physical copy of the duplicate blocks.
- When postprocess deduplication or compression operations are started on the Infinite Volume, separate operations are run on each of the data constituents up to the maximum of eight per node. If more than the maximum number of allowed compression/deduplication operations are scheduled to run at any one time, they will be queued and run as each deduplication operation completes.
- As with FlexVol volumes, a data constituent requires free space in the volume and aggregate for deduplication metadata. See the Deduplication Metadata Overhead section for more details.
- The namespace constituent is not deduplicated or compressed.
- Some commands will not operate as they do on FlexVol volumes. These include the following.
  - You can choose to view aggregated space savings for an Infinite Volume, or you can see it at the individual data constituent level. To see savings at the data constituent level, run the `volume show` command, substituting the "–volume <volname>" option with "-is-constituent true": ex. `volume show -vserver <vserver_name> -is-constituent true`.
  - The `volume efficiency show` command does not currently display information about status or progress for an Infinite Volume. To see the status or progress of individual data constituents, substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true –fields progress`.

HA Pair Controller Configuration

HA pair controller configurations are supported with both deduplication and compression. NetApp recommends that both nodes run the same version of Data ONTAP.

Deduplication and compression both run on each node independently.

Writes to the FlexVol volume or data constituents have fingerprints written to the change log. During failover or giveback to the partner node, inline compression continues as normal and deduplication change logging continues to happen. Postprocess compression and deduplication processes will start at the next scheduled time on the failed node after failover / giveback is complete, using the updated change log, or they can be started manually. A maximum of eight concurrent compression/deduplication processes are allowed on each node on an HA pair configuration. During failover, the surviving node will continue to be limited to a maximum of eight concurrent compression/deduplication operations, even though deduplication will continue for both nodes.

NetApp recommends that no active postprocess compression/deduplication operations take place during the planned takeover or planned giveback.

1. Perform the planned takeover or giveback during a time when compression/deduplication operations are not scheduled to run.
2. Determine whether any compression/deduplication operations are active and stop them until the planned takeover or giveback is complete.

You can use the `volume efficiency show` command to determine whether the status of compression and deduplication is Active or Idle. If compression/deduplication is running, then the status will be Active. There is no way to see the status or progress at the Infinite Volume level. To see the status or progress of individual data constituents, substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true –fields progress.`

On a system with compression/deduplication enabled, the output of the `volume efficiency show` command is similar to the following.

```
Vserver      Volume   State     Status     Progress

Vsone        vol1     Enabled   Active     122 GB Scanned, 25 GB Compressed

Vsone        vol2     Enabled   Active     164 GB Searched

Vsone        vol3     Enabled   Active     41 GB (45%) Done

Vsone        vol4     Enabled   Idle       Idle for 00:02:43
```

You can use the `volume efficiency stop` command to abort the active compression/deduplication operations on the volume and the `volume efficiency start` command to restart it. The `volume efficiency stop` command will create a checkpoint and, when takeover or giveback completes, you can restart the deduplication/compression operations from the latest checkpoint with the `volume efficiency start` command. For additional information about active-active controller configurations, refer to TR-3450: [Active-Active Controller Configuration Overview and Best Practice Guidelines](#).

Quotas

Quotas are based on logical space usage; therefore, for compressed and deduplicated files, the logical (uncompressed/undeduplicated) size is charged against the quotas. There are several advantages to this scheme as opposed to charging quotas based on the physical (compressed/deduplicated) size of the file.

- This is in line with the general design principle of making deduplication and compression transparent to the end user.
- It is easier for system administrators to manage quotas. They can maintain a single quota policy across all volumes, whether or not compression or deduplication is enabled on it.
- Overwriting parts of the file will not fail due to quota errors when the new data being written is not as compressible as the data it is replacing.
- Quotas cannot be oversubscribed on a volume. For example, a user with a quota limit of 1TB can't store more than 1TB of data in a deduplicated/compressed volume, even if the data has been deduplicated/compressed and fits into less than 1TB of physical space on the storage system.

FlexCache

Caching systems will not store data as compressed or deduplicated.

Flash Cache Cards

Flash Cache cards can be used to reduce the number of random disk reads of blocks that are read repeatedly. Therefore, this can be seen to improve the read performance in environments in which there are a lot of shared blocks that are read repeatedly.

The amount of performance improvement deduplication-enabled volumes will have with Flash Cache depends on the duplication rate, the access rate, the active dataset size, and the data layout. Adding a Flash Cache card to a system does not increase the deduplication maximum volume size for that system. See the Performance section for additional recommendations regarding deduplication and Flash Cache.

Deduplication will also enable efficient use of Flash Cache. It does this by retaining the deduplication savings on Flash Cache that exist on disk. In that way, if you have 32k duplicate blocks on disk, after you run deduplication only one block will be used on disk and, if it is randomly accessed, only one block will be used in Flash Cache as well. This can significantly increase the amount of data that can be stored in Flash Cache.

The Flash Cache card can provide significant performance improvements in VMware VDI environments. The advantages provided by NetApp Flash Cache cards are further enhanced when combined with other shared block technologies, such as NetApp deduplication or FlexClone. For additional information about the NetApp and VMware solution, refer to TR-3705: NetApp and VMware Solution Guide.

A volume on which compression has been run can contain both compressed and uncompressed blocks. Recall that when a compression group doesn't achieve a minimum of 25% savings or if a file is 8KB or less the blocks are written as uncompressed to the volume. How Flash Cache works with blocks depends on whether the blocks are written on disk in a compressed or uncompressed format, regardless of whether compression remains enabled or disabled. Flash Cache cards cache random reads only, it does not cache sequential reads. Since read requests for compressed data is performed sequentially any blocks on disk that contain compressed data will not be stored in the Flash Cache cards and therefore not realize any read performance improvements from the Flash Cache cards. However, the uncompressed blocks on disk and the metadata will continue to benefit from the performance improvements of Flash Cache. For additional information about the Flash Cache card, refer to TR-3832: Flash Cache Best Practice Guide.

Flash Pool

Starting in Data ONTAP 8.1.1, NetApp introduced Flash Pool. Flash Pool allows flash technology in the form of solid-state disks (SSDs) and traditional hard disk drives (HDDs) to be combined to form a single Data ONTAP aggregate. When SSD and HDD technologies are combined in a Data ONTAP aggregate, the NetApp storage system takes advantage of the latency and throughput benefits of the SSD while maintaining the mass storage capacity of HDD.

In environments with high amounts of shared blocks that are read repeatedly or overwritten randomly, Flash Pool can significantly reduce the number of HDD reads and writes, thus improving performance. If data is write cached it will still be evaluated for deduplication the next time deduplication is run. If the data is successfully deduplicated it will remain on SSD until it becomes cold and is ejected as normal. The amount of performance improvement with Flash Pool depends on the amount of shared blocks, the access rate, the active dataset size, and the data layout.

A volume on which compression has been run can contain both compressed and uncompressed blocks. Recall that when a compression group doesn't achieve a minimum of 25% savings or if a file is 8KB or less the blocks are written as uncompressed to the volume. How Flash Cache works with blocks depends on whether the blocks are written on disk in a compressed or uncompressed format, regardless of whether compression remains enabled or disabled. Flash Pool caches random reads and overwrites, it does not cache sequential I/O. Since I/O requests for compressed data is performed sequentially any blocks on disk that contain compressed data will not be stored on the SSD drives and therefore not realize any read or write performance improvements from the Flash Pool. Metadata and uncompressed data will continue to benefit from the throughput and latency improvements from Flash Pool. Which data is written to the Flash Pool will not be affected by having compression enabled. This is because any data that is randomly overwritten will not be inline compressed and therefore be written to the SSD drives. If the data is later compressed, then the uncompressed blocks will be flushed from the SSD drives and the compressed version will be written to HDD. Any data that is sequentially written will continue to be written to HDD regardless of whether compression is enabled or not.

Deduplication will also make efficient use of Flash Pool. It does this by retaining the deduplication savings on SSD that exist on HDD. In that way, if you have 32k duplicate blocks on disk, after you run deduplication only one block will be used on disk (HDD) and, if any requests for duplicate blocks are randomly requested, only one block will be used in the Flash Pool (SSD) as well. This can significantly increase the amount of data that can be stored in the Flash Pool. For more information on Flash Pool, refer to TR-4070: NetApp Flash Pool Design and Implementation Guide.

FlexClone Volumes

FlexClone technology instantly replicates data volumes and datasets as transparent virtual copies—without requiring additional storage space. FlexClone volumes support both compression and deduplication. When a FlexClone volume (cloned volume) is created:

If the parent FlexClone volume has compression or duplication enabled, the new volume inherits the compression/deduplication savings as well as the configuration of the parent volume, such as the compression/deduplication schedule.

The data in the cloned volume inherits the space savings of the original data. The deduplication process also continues for any new data written to the clone and recreates the fingerprint database in the aggregate from the volume copy of the deduplication metadata files (the fingerprint database and the change log files). This allows the deduplication process to continue to deduplicate between the new data and the old data.

It is important to understand that if you compress existing data with the `-shared-blocks true` or `-snapshot-blocks true` options on the clone, then all compressible data on the parent will be written as new data on the clone. This can significantly increase the size of the clone and thus reduce the space saving benefits of cloning.

One use case for FlexClone volumes with compression or deduplication is to quickly set up an environment with compression or deduplication to determine the possible deduplication/compression savings on a volume. This is useful in the following three scenarios when you don't want to make changes to the production environment.

- When you have a parent volume that does not already have deduplication enabled
- When you have a parent volume that does not already have deduplication or compression enabled
- When you have a parent volume that has deduplication but you want to test adding compression

Scenario 1: Adding Deduplication on a Normal Volume

To run deduplication on a FlexClone volume you run `volume efficiency start –scan-old-data` on the clone. This deduplicates the data within the clone volume only, and does not modify the data on the parent volume. To determine the savings achieved by deduplication, run the `volume show – vserver <SVMname> –volume <clonename> –dedupe-space-saved` command on the clone, as you would for a non FlexClone volume.

Scenario 2: Adding Compression and Deduplication on a Normal Volume

To add both compression and deduplication you run `volume efficiency start -scan-old-data true -vserver <SVMname> -volume <clonename > -snapshot-blocks`. After both operations complete the clone contains the compressed and deduplicated data. To determine the total savings you look at `volume show –vserver <SVMname> –volume <clonename > –sis-space-saved`. Since all compressed data is written as new data to the clone it can significantly increase the size of the clone and thus reduce the space saving benefits of cloning.

Scenario 3: Adding Compression on a Deduplicated Volume

To add compression to a deduplicated volume you enable compression on the clone and then run volume efficiency start -scan-old-data true -vserver <SVMname> -volume <clonename > -snapshot-blocks true – shared-blocks true. This results in the compressible data being written to the clone; it does not affect the data on the parent volume. To determine the savings from compression you run the volume show –vserver <SVMname> –volume <clonename > –dedupe-space-saved command. From here you can run your tests against the deduplicated/compressed clone. Since all compressed data will be written as new data to the clone it can significantly increase the size of the clone and thus reduce the space saving benefits of cloning.

All of these use cases allow you to create copies of volumes without affecting your production data. You can also choose to enable compression and/or deduplication on the clone without running with the –scan-old-data option. This will compress/deduplicate only the new data written to the volume and not show the savings from the original data in the parent.

Volume Splitting

When a cloned volume is split from the parent volume, all of the deduplicated data in the clone that was part of the parent volume (that is, not including the data that was written to the cloned volume after the clone was created) gets undeduplicated after the volume split operation. If deduplication is enabled on the cloned volume, this data gets deduplicated again in subsequent deduplication operations on the clone volume. If compression is enabled on the cloned volume, then, during a split, the data from the parent is transferred to the clone as uncompressed and will be recompressed on the clone.

FlexClone Files

FlexClone volumes at the file and LUN level are available and are allowed on compressed and/or deduplicated volumes. Both compression and deduplication can be used to regain capacity savings after a FlexClone volume is broken.

DataMotion for Volumes

DataMotion™ software for Volumes is supported with both compression and deduplication. The savings will be maintained at the destination, and deduplication and compression operations will continue as normal after the move is complete. If postprocess compression or deduplication is actively running during the move, then DataMotion will stop these operations shortly before the final cutover is complete. Compression and deduplication will continue when they are scheduled next to run. The aggregate copy of the deduplication metadata will also move; therefore, deduplication will continue on the destination after the move as normal.

If using compression, the destination aggregate must be 64-bit, and both the source and destination must run the same version of Data ONTAP or the volume move will fail.

Nondisruptive Upgrades

Both major and minor nondisruptive upgrades are supported with compression and deduplication. NetApp recommends that no active postprocess compression or deduplication operations run during the nondisruptive upgrade.

- Perform the Data ONTAP upgrade during a time when compression/deduplication operations are not scheduled to run.
- Determine whether any compression/deduplication operations are active and, if they are, halt them until the Data ONTAP upgrade is complete.

  You can use the `volume efficiency show` command to determine whether the status of compression/deduplication is Active or Idle. On a system with compression and deduplication enabled, the output of the `volume efficiency show` command is similar to the following.

```
Vserver      Volume    State      Status    Progress
Vsone        vol1      Enabled    Active    122 GB Scanned, 25 GB Compressed
Vsone        vol2      Enabled    Active    164 GB Searched
Vsone        vol3      Enabled    Active    41 GB (45%) Done
Vsone        vol4      Enabled    Idle      Idle for 00:02:43
```

- You can use the `volume efficiency stop` command to abort the active compression/deduplication operations on the volume and the `volume efficiency start` command to restart it.

For specific details and requirements for performing a nondisruptive upgrade on your system, refer to Upgrade Advisor in the AutoSupport™ tool if you have AutoSupport enabled; otherwise, refer to the release notes for the version of Data ONTAP to which you are upgrading.

32-Bit Aggregate Support

Deduplication supports both 32-bit and 64-bit aggregates. Compression is only supported with 64-bit aggregates. If you want to enable compression on a dataset that exists on a 32-bit aggregate, you must either upgrade the aggregate or migrate the data from the 32-bit aggregate to a volume in a 64-bit aggregate. When data is migrated by using a logical migration method such as NDMPcopy the data is undeduplicated during the process. Once the transfer is complete you can run deduplication again to regain the deduplication savings. For more information on upgrading an aggregate from 32-bit to 64-bit refer to TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices.

64-Bit Aggregate Support

Both compression and deduplication support 64-bit aggregates. For more information on 64-bit support refer to [TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices](#).

DUMP

Both deduplication and compression support backup to a tape via NDMP. The data sent from the source volume and written to the tape is in uncompressed/undeduplicated format.

Reallocation (realloc)

Starting in Data ONTAP 8.1 deduplicated data can be reallocated using physical reallocation or read_realloc space_optimized. Although data may be shared by multiple files when deduplicated, reallocate uses an intelligent algorithm to only reallocate the data the first time a shared block is encountered. Prior versions of Data ONTAP do not support reallocation of deduplicated data and will skip any deduplicated data encountered. Compressed data will not be reallocated by reallocate or read reallocate, and NetApp does not recommend running reallocate on compressed volumes.

Aggregate Copy Command

When deduplicated or compressed data is copied by using the `aggr copy` command, the copy of the data at the destination location inherits all of the deduplication/compression attributes and storage savings of the original data. Aggregate copy is also the only way to copy data, maintain the layout of data on disk, and enable the same performance with deduplicated data.

Vol Copy Command

When deduplicated or compressed data is copied by using the `volume copy` command, the copy of the data at the destination location inherits all of the deduplication/compression attributes and storage savings of the original data. The deduplication metadata files will automatically be recreated in the destination aggregate the next time deduplication is run.

`Vol copy` will be disallowed if the source volume is compressed with Data ONTAP 8.1 or later and the destination is running an earlier version of Data ONTAP or if the destination FlexVol volume is contained within a 32-bit aggregate. Here is the error message that gets generated: "VOLCOPY: Aborting: the aggregate that contains destination volume vol_32 does not support compression"

SnapDrive

SnapDrive® for Windows removes white space. This may reduce the expected savings from compression or deduplication; however, the overall system savings will still benefit from the effects of SnapDrive capacity reduction. The effects on the overall space savings are not significant, and will usually not warrant any special actions.

LUNs

When using deduplication or compression on a file-based (NFS/CIFS) environment, both deduplication and compression are straightforward and automatic. As data is written, it is compressed inline or postprocess. As duplicate blocks are freed from deduplication, they are marked as available. In both cases, as blocks of free space become available, the NetApp system recognizes these free blocks and makes them available to the volume.

Deduplication and compression on a block-based (FCP/iSCSI) LUN environment are slightly more complicated. This is because of the space guarantees and fractional reservations used by LUNs. For instance, consider a volume that contains a 500GB LUN, and the LUN has LUN reserves enabled. The LUN reserve causes the space for the LUN to be reserved when the LUN is created. Now consider that 500GB of data is written to the LUN. The consumed space is exactly 500GB of physical disk space. If the

data in the LUN is reduced through compression or deduplication, the LUN still reserves the same physical space capacity of 500GB, and the space savings are not apparent to the user.

LUN space guarantees and fractional reserves can be configured so that the use by the NetApp system of the freed blocks changes depending on the configuration. By varying the values of certain parameters, freed blocks can be returned to either the volume free pool or the aggregate free pool.

This section describes four common examples of LUN configurations and the associated compression and deduplication behavior, as summarized in the table below.

Table 14) Summary of LUN configuration examples.

	A (Default)	B	C	D
LUN space guarantee value	Yes	Yes	No	No
Volume fractional reserve value	100	0	0	0
LUN (fractional) overwrite reserved	Yes	Yes	No	No
Volume space guaranteed	Yes	Yes	Yes	No
Compression and deduplication savings are returned to which free pool?	Neither	Neither	Volume free pool	Aggregate free pool

Definitions

LUN (Fractional) overwrite reserve: The space that Data ONTAP reserves will be available for overwriting blocks in a LUN when the space guarantee = Yes. Since this space is reserved, any savings from compression or deduplication will not be available for other use.

Volume free pool: Refers to the free blocks in the parent volume of the LUN. These blocks can be assigned anywhere in the volume as needed.

Aggregate free pool: Refers to the free blocks in the parent aggregate of the LUN. These blocks can be assigned anywhere in the aggregate as needed.

Configuration A: Default LUN Configuration

The default configuration of a NetApp LUN follows. (The best practice for all NetApp LUNs is to turn the controller Snapshot copy off, delete all scheduled Snapshot copies, and set the snap reserve to 0.)

LUN space reservation value = on	Default
Volume fractional reserve value = 100	Default
Volume guarantee = volume	Default
Snap reserve = 0% (Default = 5%)	`vol modify -percent-snapshot-space 0`
Autodelete = off	Default
Autosize = off	Default
Try_first = volume_grow	Default

This configuration is used so that overwrites to the LUN never fail, even if it is overwritten entirely. When a LUN is created with LUN space reservation on and volume guarantee set to volume, no apparent compression or deduplication savings are observed by the storage administrator.

Configuration B: LUN Configuration with Volume Free Space Management

If the user wants to remove LUN overwrite reserve space, this can be accomplished with the following configuration.

LUN space reservation value = on   Default

Volume fractional reserve value = 0   `vol modify -fractional-reserve 0`

Volume guarantee = volume   Default

Snap reserve = 0%   `vol modify -percent-snapshot-space 0`

Autodelete = on   `snapshot autodelete modify –enabled true`

Autosize = on   `vol modify –max-autosize X` –autosize-increment Y
where X represents the maximum size to which the volume can grow and Y represents the increment by which to grow the volume; both are set in values of `<size>[k|m|g|t]`)

Try_first = volume_grow   Default

The differences between this configuration and configuration A is that the value of the fractional reserve is set to zero and both autodelete and autosize are on. As a result, in this configuration no blocks are set aside for LUN overwrites. To mitigate the possibility of overwrite failures caused by a full volume, NetApp recommends turning on snapshot autodelete and volume autosize. Snapshot autodelete will free space in a volume by automatically deleting old snapshots, and volume autosize will expand the volume when it reaches a predefined threshold.

In this configuration, if compression or deduplication was run, there would be no space savings even if a space reclamation tool was run. The reason is that with LUN space reservation on, 100% of the LUN space is allocated at the time of LUN creation whether those blocks contain data or are empty.

Configuration C: LUN Configuration for Maximum Volume Space Savings

If the user wants to make freed blocks available to the volume free pool, this can be accomplished with the following configuration.

LUN space reservation value = off   `lun modify -space-reserve disabled lun_path`

Volume fractional reserve value = 0   `vol modify -fractional-reserve 0`

Volume guarantee = volume   Default

Snap reserve = 0% (Default = 5%)   `vol modify -percent-snapshot-space 0`

Autodelete = on   `snapshot autodelete modify –enabled true`

Autosize = on   `vol modify –max-autosize X` –autosize-increment Y
where X represents the maximum size to which the volume can grow and Y represents the increment by which to grow the volume; both are set in values of `<size>[k|m|g|t]`)

Try_first = volume_grow   Default

The difference between this configuration and configuration B is that the LUN is not space reserved. This means that at the time of creation the LUN takes up no space from the volume. Only when data is written to the LUN is space allocated. This allows volumes to support more LUNs than they physically have space for. With LUN space guarantees off, the value for the volume fractional reserve is ignored for all LUNs in this volume. However, since autodelete and autosize are both on, the volume will expand rather than run out of space if the total allocated blocks approach the total size of the volume.

In this configuration, after compression/deduplication is run, the savings will not be seen unless a space reclamation tool such as SnapDrive is run. If space reclamation is run after compression/deduplication completes, then the freed blocks will be made available to the volume.

Configuration D: LUN Configuration for Maximum Volume and Aggregate Space Savings

This configuration provides the optimal space savings by returning all freed blocks to the aggregate free pool. This is accomplished with the following configuration:

LUN space reservation value = off	`lun modify -space-reserve disabled lun_path`
Volume fractional reserve value = 0	`vol modify -fractional-reserve 0`
Volume guarantee = none	`vol modify -space-guarantee none`
Snap reserve = 0% (Default = 5%)	`vol modify -percent-snapshot-space 0`
Autodelete = on	`snapshot autodelete modify –enabled true`
Autosize = on	`vol modify –max-autosize X –autosize-increment Y`

where X represents the maximum size to which the volume can grow and Y represents the increment by which to grow the volume; both are set in values of `<size>[k|m|g|t]`)

Try_first = volume_grow	Default

The difference between this configuration and configuration C is that the volume is not space reserved. This configuration automatically allocates the free blocks from the volume into the aggregate free pool, where the blocks can be reallocated to any other volumes or LUNs in the aggregate.

The advantage of this configuration is that it provides the highest efficiency in aggregate space provisioning. It also uses the thin provisioning features of Data ONTAP, volume autosize, and Snapshot autodelete to help administer the space in the solution.

In this configuration any blocks in the volume that are freed by compression and deduplication are automatically allocated to the aggregate free space pool. If space reclamation is performed on the LUN, then any freed blocks from the LUN will also be allocated to the aggregate free pool.

For more information on LUN configuration options, see TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide.

15 Troubleshooting

This section discusses basic troubleshooting methods and common considerations when working with compression and deduplication.

15.1 Maximum Logical Data Size Processing Limits

In Data ONTAP 8.1, the maximum logical data size that will be processed by postprocess compression and deduplication is equal to the maximum volume size supported on the storage system regardless of the size of the volume or data constituent created. Starting in Data ONTAP 8.2, the maximum logical data size that will be processed by postprocess compression and deduplication is 640TB regardless of the size of the volume created. If you have surpassed this logical limit, postprocess compression and deduplication operations will fail with the error message "maximum logical data limit has reached." This could result in lower than expected savings on volumes or data constituents with large amounts of data.

15.2 Too Much Impact from Inline Compression and Not Much Savings

There are many factors that could be involved if you find that the impact with inline compression is too great. One possible reason is that your volumes contain a mix of highly and minimally compressible data. To offset this impact, starting in Data ONTAP 8.2 we added the idd option (incompressible data detection) to detect and skip inline compression of files with little or no savings while still getting immediate savings from inline compression on other files that have greater savings. Since you still want to get the maximum possible savings from all your files, you still compress the files skipped by inline compression at the next scheduled run of postprocess compression.

The way it works depends on whether you are writing a large or a small file. By default we assume a file that is less than 500MB is a small file. You can modify the size of what is considered a small or a large file by modifying the `quick-check-fsize` option of the `volume efficiency config` command.

For small files we will compress each compression group within the file unless or until we find one that has less than 25% savings. At that time we will mark the file as incompressible and stop any additional inline compression attempts. For large files we will try to compress the first 4K of each compression group. If there is at least 25% savings in the first 4K we will continue compressing the rest of the compression group. If the first 4K of a compression group does have at least 25% savings we will not continue inline compression attempts on the rest of the compression group and will write it as uncompressed. We will continue to do this quick check on all compression groups within the file.

Postprocess compression will continue to try to compress any compression groups that were skipped by inline compression.

To determine if you might benefit from using the idd option, check the following two counters from the `volume efficiency stat` output:

```
Inline Compression Attempts :  Total number of inline compression attempted

Inline Incompressible CGs   :  Total number of incompressible CG found by
inline compression
```

This should be checked over time. If you see that the ratio of incompressible CG to inline compression attempts is =>0.8 over a period of time, consider setting this option to true. By default, this option is set to false.

15.3 Postprocess Operations Taking Too Long to Complete

When compression and deduplication begin, compression will run first, followed by deduplication. Compression will read the uncompressed blocks into memory, compress the data, and rewrite the blocks as compressed. When the deduplication process begins it will sort and search the fingerprint database, and then remove all verified duplicates. This process can be timely, depending on the amount of data to be processed. In some cases other factors may play a key role, and should be considered accordingly.

When running compression or deduplication, verify that there are enough resources available. Another consideration is the number of simultaneous compression or deduplication processes that are running on a single system. Follow the best practice of using a gradual approach to determine how many simultaneous compression and deduplication processes you can run safely in your environment before saturating your system resources.

A common practice is to run compression and deduplication at off-peak hours. If compression or deduplication does not complete during off-peak hours, consider stopping the compression or deduplication processes during peak hours and resuming them during off-peak hours. When you stop compression and deduplication processes a checkpoint will automatically be created that will be used the next time you start the processes.

NetApp strongly recommends that performance testing be done prior to using deduplication or compression in your environment.

15.4 Lower-Than-Expected Space Savings

If you do not see the expected savings when using deduplication or compression, consider the following factors.

There May Not Be Enough Space for Deduplication to Run

The aggregate should have 3% of the total data's worth of free space for all deduplicated FlexVol volumes or data constituents, and each FlexVol volume or data constituent should have 4% of the total data's worth of free space. For additional details about the overhead associated with the deduplication metadata files, see the section above on [Deduplication Metadata Overhead](#).

The LUN Configuration May Be Masking the Deduplication Savings

Different LUN configurations will cause freed blocks to be returned to different logical pools within the storage system. Refer to the LUN section in "[Feature Interoperability](#)," above, for information about deduplication with LUNs and how space savings are recognized with different configurations.

Snapshot Copies

Snapshot copies lock blocks in place by design, so that they are available for data recovery. This locking mechanism does not allow blocks that are freed by deduplication or rewritten with compression to be returned to the free pool until the locks expire or are deleted. If you are using deduplication or compression with Snapshot copies, consider the following.

- If possible, run compression and deduplication to completion before taking a Snapshot copy.
- Use the `snap show` command to see what Snapshot copies exist and the `snap delete` command to remove them. Alternatively, wait for the Snapshot copies to expire and the space savings to appear.
- By default the initial compression of existing data skips blocks that are locked in Snapshot copies. If your existing data on disk contains a lot of data that is locked by Snapshot copies, running compression with the `-scan-old-data` without the `-snapshot-blocks true` option may not show much savings.

For more information on the compression of existing data and its options see the [Compression and Deduplication Options](#) section, above. For more information about using compression and deduplication with Snapshot copies, see [Snapshot Copies](#) in the "Feature Interoperability" section, above.

There May Be Little Duplicate or Compressible Data Within the Volume

Run the Space Savings Estimation Tool (SSET) against the dataset to get an idea of the amount of duplicate and compressible data within the dataset.

Compression of Existing Data on Disk Runs on Already Deduplicated Data

By default the initial compression of existing data on disk skips blocks that are deduplicated. If the existing data on disk contains a lot of deduplicated data, compressing the existing data without the `-shared-blocks` option may not result in much savings.

Misaligned Data Can Result in Reduced or Little Space Savings

This issue is a bit harder to track down. If you believe this may be the issue, it is best to contact [NetApp Customer Success Services](#) for assistance.

Misaligned blocks are a possibility if best practices are not followed when configuring a virtualization environment. For more information on VMware best practices, see [TR-3749: NetApp and VMware vSphere Storage Best Practices](#).

Snap Reserve Space Is Exceeded

Once the snapshot reserve space is full, Snapshot copies will begin to use space in the active file system. The `df -S` command reports savings by dividing 'saved' by 'used + saved' reported in the active file system only. The savings value does not take into account space used by Snapshot copies regardless of whether they include savings. This means that if Snapshot copies start taking space from the active file system, this space will be counted as used space in the active file system and therefore will decrease the value reported as %saved in the `df -S` output. Consider the following example:

1. Create a new volume with the snapshot reserve set to 0, populate it with data, and take a Snapshot copy.

```
Cluster1::> volume show -volume test -fields percent-snapshot-space
vserver volume   percent-snapshot-space
-------  ----------  --------------------------------
vsone   test      0%
```

```
Cluster1::> snap show -vserver vsone -volume test
                                         ---Blocks---
vserver Volume Snapshot                 Size    Total%  Used%
-------- ------------ --------------------------------- --------- ---------- -----
vsone   test       daily.2011-08-15_0010  220KB   0%      3%
```

```
Cluster1::> df -S test
File system        used      total-saved   %total-saved   deduplicated   %deduplicated   compressed
%compressed  Vserver
/vol/test/         185584732     0            0%            0               0%              0
0%              vsone
```

```
Cluster1::> df -h test
File system        total     used      avail   capacity  Mounted on      Vserver
/vol/test/          500GB    176GB     323GB    35%    /vol/test/        vsone
/vol/test/.snapshot   0KB    5884KB    5318GB   ---%    /vol/test/.snapshot   vsone
```

2. Run compression against the existing data with the option to compress data already locked in a Snapshot copy. (This must be run in advanced mode.)

```
Cluster1::> set advanced
Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel. Do you want to continue? {y|n}: y
```

```
Cluster1::*> volume efficiency start -vserver <SVMname> -volume test - scan-old-
data true -compression true -shared-blocks true
```

3. After compression completes notice the amount of Snapshot copy space used and the savings results from `df -S`.

```
Cluster1::> df -S test
File system        used      total-saved   %total-saved   deduplicated   %deduplicated   compressed
%compressed  Vserver
/vol/test/         264527140   93001804      26%           0               0%              93001804
26%             vsone
```

```
Cluster1::> df -h test
File system              total      used      avail  capacity  Mounted on
/vol/test/              500GB      252GB     247GB    50%      /vol/test/
/vol/test/.snapshot     0KB       163GB     5243GB   ---%     /vol/test/.snapshot
```

4. Change the snapshot reserve space so that it is large enough to contain the full size of the Snapshot copy. Notice the savings results from `df -S`.

```
Cluster1::> volume show -volume test -fields snapshot-space-used
vserver         volume         snapshot-space-used
-------         ----------     -----------------------------
vsone           test           40%
```

```
Cluster1::> df -S test
File system         used     total-saved   %total-saved   deduplicated   %deduplicated   compressed
%compressed  Vserver
/vol/test/          92598792  93001804       50%            0              0%              93001804
50%           vsone
```

```
Cluster1::> df -h test
File system              total      used      avail  capacity  Mounted on
/vol/test/              300GB      88GB      111GB    44%      /vol/test/
/vol/test/.snapshot     200GB     163GB     36GB     82%      /vol/test/.snapshot
```

Here's why, savings is calculated by dividing 'used space' by 'used + savings.'

Snapshot size: 163*1024*1024 = 170917888

Used: 264527140

Saved: 93001804

Savings equals: saved / (used + saved))
              93001804 / (264527140+93001804) = 26%

To calculate the actual savings in the active file system (space not including Snapshot copies), you would subtract the amount of space used by the Snapshot copies that exceeds the snapshot reserve space from the value listed as "Used" in the calculation.

Active File system Space: 93609252 [used - snapshot size (264527140 – 170917888)]

Savings equals: saved / (used + saved)
              93001804 / (93609252+93001804) = 50%

This 50% savings represents the actual savings you will realize when the Snapshot copies containing the original uncompressed blocks are deleted or expire.

15.5 Slower-Than-Expected Performance

The information in this section is provided assuming that basic proof-of-concept testing has been performed prior to running in production to understand what performance to expect when compression and deduplication are enabled on the NetApp system. NetApp strongly recommends that performance testing be performed prior to using deduplication or compression in your environment. If unexpectedly slow performance continues to be an issue, NetApp highly recommends that NetApp Customer Success Services be contacted for in-depth troubleshooting. In many cases there are other factors such as misconfigured applications or conflicting policies that can be easily fixed to regain acceptable performance.

Unexpectedly Slow Read Performance Caused by Adding Deduplication

Deduplication can affect sequential read performance: The more deduplication savings, the more effect is possible. Some examples of applications that use sequential reads are NDMP, restore, and some NFS apps. Also, if the data is being read in such a way that the deduplicated data cannot be maintained within the system memory, there may be a performance impact. To offset this effect, NetApp provides intelligent caching. Intelligent caching provides the ability to cache deduplicated blocks, resulting in a performance boost for read requests. Intelligent caching applies to Flash Cache and Flash Pools. Refer to the "Feature Interoperability" section for information about compression and deduplication with Flash Cache cards and Flash Pool.

In the case of random reads, there is usually not much if any impact on performance from random reads of deduplicated data.

Unexpectedly Slow Read Performance Caused by Adding Compression

Compression can affect read performance due to the necessity of uncompressing compression groups before servicing a read. This effect is minimized due to the size of the compression group but will have the most effect on small random read performance. The least effect will be felt on large sequential reads.

Unexpectedly Slow Write Performance Caused by Adding Compression or Deduplication

Enabling deduplication on a volume will cause the creation of deduplication metadata (fingerprints) as data is written to the volume. The deduplication metadata is a standard part of the deduplication process. The creation of the metadata is not typically an issue on systems that have available resources for the deduplication process; however, if the storage system resources are used by other applications, then even the slightest impact can have a visible impact on the storage system.

If write performance appears to be degraded, check the NetApp system resources (CPU, memory, and I/O) to determine that they are not saturated. If resources are saturated you can consider stopping some or all compression and deduplication operations to see if performance resumes. Stopping compression and deduplication operations will generate a checkpoint and these operations can be resumed at a time when the system is less busy. If system resources are still saturated, you can consider disabling inline compression if enabled and see if resource usage levels drop sufficiently. Remember, turning off compression and deduplication will not undo compressed and deduplicated blocks already on disk. Disabling compression and deduplication will stop the creation of deduplication metadata, stop any future postprocess compression and deduplication from running, and stop new writes from being inline compressed. All compression and deduplication savings achieved prior to that point will continue to exist. Also, any reads and partial overwrites will still require uncompression of the compression group in memory.

It is also worth noting that write performance will vary based on different platforms. So it is not typically a good approach to compare write performance results across different NetApp platforms. Write performance can also be affected by using slower disk drives. Thus, comparing FC drives to SATA drives can give different results.

If slow write performance continues to be an issue, NetApp highly recommends that NetApp Customer Success Services be contacted for in-depth troubleshooting. In many cases there are other factors such as misconfigured applications or conflicting policies that can be easily fixed to regain acceptable performance. See the "Contact Information for NetApp Customer Success Services" section of this document for contact information and data collecting guidance.

System Running Slower Since Enabling Compression/Deduplication

Deduplication has been known to compound system performance issues associated with misalignment. For instance, a misconfiguration could be causing the NetApp system to run at less than optimum performance levels without being noticed. A good example is the case of misaligned blocks in a VMware ESX® environment. See [TR-3749: NetApp and VMware vSphere Storage Best Practices](#), for additional details. When deduplication is enabled in this environment, it is common to see noticeable degradation in system performance, because deduplication will effectively cause the misconfigured blocks to be accessed more often since they are now shared and more likely to be accessed. In this case, other troubleshooting efforts outlined within this document will not resolve this issue. The best approach for this scenario is to contact NetApp Customer Success Services. The support team will be able to analyze the system and recommend a remedy to resolve the performance issue while maintaining the space savings.

Another common performance caveat occurs when too many compression/deduplication processes are run simultaneously on a single system. The maximum number of simultaneous compression/deduplication processes that can be run on a single storage system is eight. Running eight simultaneous compression/deduplication processes will use significant resources, and can reduce application performance. The number of simultaneous compression/deduplication processes that will run on a system should be reevaluated as additional applications and processes are run on the storage systems. If need be consider running fewer compression/deduplication processes simultaneously. The best approach is to rerun the performance testing on the system to understand how compression/deduplication will run once the additional workload is added to it. This performance testing would likely entail reducing the number of simultaneous compression/deduplication processes to one for a better understanding of the performance effects, and then systematically increasing the number of simultaneous compression/deduplication processes until a system performance threshold is reached.

Although it is relatively easy to uncompress or undeduplicate a FlexVol volume, the process is resource intensive and can take a significant amount of time to complete. As an alternative, you can disable compression and deduplication. This will stop any future compression or deduplication from occurring. All space savings from compression and deduplication run prior to their being disabled will continue to exist. Undoing compression or deduplication should be considered only if recommended by NetApp Customer Success Services, following their analysis of the environment to ascertain whether the source of the problem is being properly addressed. If you must undo deduplication or compression, you can do it while the volume is online, as described the [Removing Space Savings](#) section, below.

15.6 Removing Space Savings

NetApp recommends contacting NetApp Customer Success Services prior to undoing deduplication or compression on a volume to determine that removing the space savings is really necessary. In many cases system performance can be restored by finding the true source of the degradation, which often can be unrelated to compression or deduplication.

It is relatively easy to uncompress or undeduplicate a volume and turn it back into a regular volume. This can be done while the volume is online, as described in this section.

Undo operations can take a long time to complete. The time to complete is relative to the amount of savings and the amount of available system resources. You can view the progress of undo operations by using the `volume efficiency show –fields` progress command. During the time when undo operations are running, there might be an impact on other activity on the system. Although undo operations are low-priority background processes, they do consume CPU, memory, and disk resources, and therefore they should be scheduled to run during low-usage times. We do not limit the number of undo operations you can run in parallel; however, the more that run, the more impact will be seen. If you determine that the impact is too great with so many in parallel or during peak times, you can stop the operation using the `volume efficiency stop` command, which will stop the current undo. You can later restart the operation, and it will continue from where it stopped.

Note: Undo operations will only remove savings from the active file system, not within Snapshot copies.

Uncompressing a Volume

Before attempting to uncompress data in a volume, verify that you have sufficient space for the increased amount of space necessary to store the data on disk. This includes not only the increase in the amount of space used in the active file system, but it might temporarily affect the amount of space used by Snapshot copies. This is because uncompression rewrites the uncompressed version as new blocks on disk. If these original compressed blocks are locked in a Snapshot copy, then both the original and newly uncompressed blocks will temporarily exist on disk until the Snapshot copy expires or is deleted.

The estimated amount of additional space you will need on disk to successfully complete undo operations is equal to the amount shown as "sis-space-saved" in `volume show` prior to the undo. Also, if you use Snapshot copies, there might be extra space used in the Snapshot copies equal to the amount of space previously saved on disk. For example, a volume with 50% compression savings with Snapshot copies will need that 50% free space for the active file system plus the additional space needed by the Snapshot copies to keep the "original" compressed blocks.

Some best practices to achieve the maximum amount of uncompression on a volume containing Snapshot copies are:

* Delete as many Snapshot copies as possible before undoing compression.
* Change the Snapshot policy to none until uncompression completes.
* If possible, temporarily reduce the retention duration of Snapshot copies until all the Snapshot copies containing compressed data expire.
* If this is a source for a backup or DR, it might cause a large number of blocks to be transferred on the next scheduled transfer since all uncompressed blocks are considered new blocks.

Note: If you restore from a Snapshot copy that contains compressed blocks, the blocks will remain as compressed in the active file system even if you already uncompressed the volume. To uncompress these blocks, you have to rerun the uncompression operation on the volume.

To remove the compression savings from a volume you must first turn off compression on the volume. To do this use the command:

```
volume efficiency modify –vserver <SVMname>  –volume <volname> –compression
false –inline-compression false
```

This command stops both inline and postprocess compression from compressing new writes to the volume. It is still possible to read compressed data while compression is turned off. This command will not undo the compression of the data already compressed on disk, and the savings will not be lost.

If compression is turned off on a volume for a period of time and then turned back on for this same volume, new writes will be compressed. Compression of existing data can be used to compress the data that was written during the period that compression was turned off.

If you want to uncompress the compressed data in a volume, after compression has been turned off use the following command:

```
volume efficiency undo –vserver <SVMname> –volume <volname> -compression true
```

Note: You must be in advanced mode to run this command.

Here is an example of uncompressing a volume:

Cluster1::> `df –S`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed %compressed	Vserver
/vol/ volHomeC/	138787476	36513720	21%	0	0%	36513720 21%	vsone

Cluster1::> `volume efficiency show -vserver vsone -volume volHomeC -fields compression, inline-compression`

(volume efficiency show)

vserver	volume	compression	inline-compression
vsone	volHomeC	true	true

Cluster1::> `volume efficiency modify -vserver vsone -volume volHomeC -compression false -inline-compression false`

(volume efficiency modify)

Cluster1::> `volume efficiency show -vserver vsone -volume volHomeC -fields compression, inline-compression`

(volume efficiency show)

vserver	volume	compression	inline-compression
vsone	volHomeC	false	false

Cluster1::> `set advanced`

Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.

Do you want to continue? {y|n}: `y`

Cluster1::*> `volume efficiency undo -vserver vsone -volume volHomeC -compression true`

(volume efficiency undo)

Sufficient free space (estimated 35GB) on volume volHomeC of Vserver vsone is available for the "volume efficiency undo" operation.

Warning: Adding more data while the "volume efficiency undo" operation is in progress will cause the undo to fail if there is insufficient free space to complete the operation. The volume efficiency undo operation for volume ' volHomeC ' of Vserver 'vsone' has started.

Cluster1::*> `volume efficiency show -volume volHomeC`

(volume efficiency show)

Vserver	Volume	State	Status	Progress
vsone	volHomeC	Enabled	Undoing	62 GB Processed

Cluster1::*> `volume efficiency show -volume volHomeC`

(volume efficiency show)

Vserver	Volume	State	Status	Progress
vsone	volHomeC	Enabled	Idle	Idle for 04:52:04

Cluster1::*> `df -S /vol/volHomeC`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed	%compressed	Vserver
/vol/volHomeC/	195661760	0	0%	0	0%	0	0%	vsone

Note: If at any time `volume efficiency undo` determines that there is not enough space to uncompress, it stops and sends a message to the console about insufficient space, and leaves the volume compressed. Use `df -r` to find out how much free space you really have, and then delete either data or Snapshot copies to provide the needed free space.

Note: Deduplication savings can be temporarily decreased after running `volume efficiency undo -compression`. This is because `volume efficiency undo -compression` will rewrite compressed blocks as new uncompressed blocks including those that previously included block sharing (that is, deduplicated or FlexClone blocks). To regain these savings you can rerun deduplication after uncompression completes.

Undeduplicating a Volume

To remove deduplication from a volume you must first turn off deduplication (this will also turn off compression if it was enabled) on the volume. To do this use the command:

```
volume efficiency off -vserver <SVMname> -volume <volname>
```

This command stops fingerprints from being written to the change log as new data is written to the volume.

Considerations:

- When you turn off deduplication (`volume efficiency off`) on a volume you also turn off both inline and postprocess compression. To run only inline compression, `deduplication and postprocess compression` needs to be enabled on the volume but the postprocess compression/deduplication schedule can be set to never run (`sis modify -schedule - - vserver <SVMname> -volume <volname>`).
- There is no way to stop postprocess deduplication and continue doing postprocess compression.
- It is possible to remove only the deduplication savings and not the compression savings.
- If this command is used and then deduplication is turned back on for this volume, the volume can be rescanned to get savings on the existing undeduplicated data with the `volume efficiency start -scan-old-data` command.

Next, if you want to remove the deduplication savings by recreating the duplicate blocks in the volume, use the following command (the `volume efficiency undo` command is available only in advanced mode):

```
set advanced
```

```
volume efficiency undo –vserver <SVMname> -volume <volname> -dedupe true
```

This command will recreate the duplicate blocks and delete the fingerprint and change log files.

Here is an example of undeduplicating a FlexVol volume:

Cluster1::> `df -S volHomeD`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed
% compressed	Vserver					
/vol/volHomeD/	81043124	94335648	54%	94335648	54%	0
0%	vsone					

Cluster1::> `volume efficiency show -volume volHomeD`

vserver	volume	State	Status	Progress
---------	----------	-------	----------	-----------------
vsone	volHomeD	Enabled	Idle	Idle for 01:35:42

Cluster1::> `volume efficiency off –vserver vsone –volume volHomeD`

   (volume efficiency off)

Efficiency for volume ' volHomeD ' of Vserver 'vsone' is disabled.

Cluster1::> `set advanced`

Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel. Do you want to continue? {y|n}: `y`

Cluster1::*> `volume efficiency undo -vserver vsone -volume volHomeD -dedupe true`

   (volume efficiency undo)

Sufficient free space (estimated 90GB) on volume volHomeD of Vserver vsone is available for the "volume efficiency undo" operation.

Warning: Adding more data while the "volume efficiency undo" operation is in progress will cause the undo to fail if there is insufficient free space to complete the operation.

The efficiency undo operation for volume ' volHomeD ' of Vserver 'vsone' has started.

Cluster1::*> `volume efficiency show -volume volHomeD`

vserver	volume	State	Status	Progress
---------	----------	-------	----------	-----------------
vsone	volHomeD	Disabled	Undoing	128 GB Processed

Cluster1::*> `volume efficiency show –volume volHomeD`

   (volume efficiency show)

There are no entries matching your query.

Cluster1::*> `df –S volHomeD`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated
compressed	%compressed	Vserver			
/vol/volHomeD/	174037920	0	0%	0	0%
0	0%	vsone			

Note: If `volume efficiency undo` starts processing and then there is not enough space to undeduplicate, it stops, sends a message about insufficient space, and leaves the volume deduplicated. Use `df –r` to find out how much free space you really have, and then remove some data from the volume or delete some Snapshot copies to provide the needed free space.

Note: Compression savings can be decreased after running `volume efficiency undo–dedupe true.` This is because `volume efficiency undo –dedupe true` will rewrite previously shared blocks as uncompressed blocks. If you want to regain these compression savings you reenable compression and deduplication. Next, you set the compression/deduplication schedule to never run, followed by running compression against the existing data. Optionally you can also enable inline compression at this point.

Uncompressing and Undeduplicating a Volume

Before attempting to uncompress and undeduplicate data in a volume, verify that you have sufficient space for the increased amount of space necessary to store the data on disk. This includes not only the increase in the amount of space used in the active file system, but it might temporarily affect the amount of space used by Snapshot copies. This is because uncompression rewrites the uncompressed version as new blocks on disk. If these original compressed blocks are locked in a Snapshot copy, then both the original and newly uncompressed blocks will temporarily exist on disk until the Snapshot copy expires or is deleted.

The estimated amount of additional space you will need on disk to successfully complete undo operations is equal to the amount shown as "sis-space-saved" column in volume show prior to the undo. Also, if using Snapshot copies, there might be extra space used in the Snapshot copies equal to the amount of space previously saved on disk. For example, a volume with 50% compression savings with Snapshot copies will need that 50% free space for the active file system plus the additional space needed by the Snapshot copies to keep the "original" compressed blocks.

Some best practices to achieve the maximum amount of uncompression and undeduplication on a volume containing Snapshot copies are:

- Delete as many Snapshot copies as possible before undoing compression and deduplication.
- Change the Snapshot policy to none until undo operations complete on the volume.
- If possible, temporarily reduce the retention duration of Snapshot copies until all the Snapshot copies containing compressed data expire.
- If this is a source for a backup or DR, it might cause a large number of blocks to be transferred on the next scheduled transfer since all uncompressed blocks are considered new blocks.

Note: If you restore from a Snapshot copy that contains compressed blocks, the blocks will remain as compressed in the active file system even if you already uncompressed the volume. To uncompress these blocks, you have to rerun the uncompression operation on the volume.

To remove both compression and deduplication savings from a volume you must first turn off deduplication and compression on the volume. To do this use the command:

```
volume efficiency off –vserver <SVMname> -volume <volname>
```

This command stops new data that is written to the volume from being inline compressed, postprocess compression and deduplication from running, and its fingerprints from being written to the change log. For more details refer to the two sections above: "Undeduplicating a Volume" and "Uncompressing a Volume."

Use the following command to uncompress and remove deduplication savings by rewriting the data in its uncompressed state and recreating the duplicate blocks (the `volume efficiency undo` command is available only in advanced mode):

```
volume efficiency undo –vserver <SVMname> -volume <volname> -dedupe true –
compression true
```

Here is an example of uncompressing and undeduplicating a volume:

Cluster1::> `df -S volHomeExisting`

File system	used	total-saved	%total-saved	Deduplicated	%deduplicated compressed	%compressed	Vserver
/vol/volHomeExisting/	72005228	103364200	59%	64621288	47% 38742912	35%	vsone

Cluster1::> `volume efficiency show -vserver vsone -volume volHomeExisting -fields state, compression, inline-compression`

  (volume efficiency show)

vserver	volume	state	compression	inline-compression
----------	----------	-------	-----------	------------------
vsone	volHomeExisting	Enabled	true	true

Cluster1::> `volume efficiency off –vserver vsone –volume volHomeExisting`

  (volume efficiency off)

Efficiency for volume ' volHomeExisting ' of Vserver 'vsone' is disabled.

Cluster1::> `volume efficiency show -vserver vsone -volume volHomeExisting -fields state, compression, inline-compression`

  (volume efficiency show)

vserver	volume	state	compression	inline-compression
----------	----------	-------	-----------	------------------
vsone	volHomeExisting	Disabled	false	false

Cluster1::> `set advanced`

Warning: These advanced commands are potentially dangerous; use them only when directed to do so by NetApp personnel.

Do you want to continue? {y|n}: `y`

Cluster1::*> `volume efficiency undo -vserver vsone -volume volHomeExisting -compression true -dedupe true`

   (volume efficiency undo)

Sufficient free space (estimated 99GB) on volume volHomeExisting of Vserver vsone is available for the "volume efficiency undo" operation.

Warning: Adding more data while the "volume efficiency undo" operation is in progress will cause the undo to fail if there is insufficient free space to complete the operation. The volume efficiency undo operation for volume ' volHomeExisting ' of Vserver 'vsone' has started.

Cluster1::*> `volume efficiency show -volume volHomeExisting`

vserver	volume	state	compression	inline-compression
vsone	volHomeExisting	Disabled	Undoing	102 GB Processed

Cluster1::*> `volume efficiency show -volume volHomeExisting`

   (volume efficiency show)

There are no entries matching your query.

Cluster1::*> `df –S volHomeExisting`

File system	used	total-saved	%total-saved	deduplicated	%deduplicated	compressed	%compressed
/vol/volHomeExisting/	174022932	0	0%	0	0%	0	0%

15.7 Logs and Error Messages

This section provides guidance for collecting system information for compression and deduplication. This information can be very useful for troubleshooting.

Interpreting Compression and Deduplication Error Messages

This section describes some special error conditions that are logged and related to compression or deduplication.

Table 15) Data compression- and deduplication-related error messages.

Error Message	Explanation	Resolution
wafl_compress_enable_failed	An attempt to enable compression on an unsupported platform. This could be caused by an attempt to enable compression on a: Traditional volume; compression is only supported on FlexVol volumes FlexVol volume that is contained within a 32-bit aggregate; compression is only supported on 64-bit aggregates Volume that already has compression enabled	Take the applicable action to either: Upgrade aggregate to 64-bit. Check if compression is already enabled on the volume.
wafl_compress_fail_rootVol	Attempt to enable compression on the root volume.	Compression cannot be enabled on the root volume.
sis_cfg_invalidEntry	SIS operation found an invalid configuration for the volume.	Default configuration is created automatically. Use `volume efficiency` command to change default settings.
sis_cfg_deleted	The deduplication configuration file has been deleted.	Reenable and reconfigure deduplication for the volume using `volume efficiency on` and `volume efficiency modify` to resume deduplication.
sis_cfg_setFailed	Deduplication failed to write in the configuration file. This can lead to incorrect reporting of statistics.	Determine that there is adequate space in the volume.
sis_ chkpoint_restore_failed	The sis operation failed to restore from the previous checkpoint.	The sis operation automatically starts from the beginning.
sis_undo_nospace	Volume does not have enough space to complete the volume efficiency undo operation.	Free up space on the volume, and then try again.
sis_change log_full	The change logging metafile is full.	Run `volume efficiency start` to process entries in the change logging metafile.
sis_aggregate_full	The aggregate is full and the change logging metafile cannot hold any more fingerprint entries.	Add disks to the aggregate or free up space to allow change logging to proceed.
sis_aggr_metadir_create_failed	The creation of deduplication metafiles in the aggregate containing the deduplication volume failed. Change logging will be turned off on the volume.	Determine that there is adequate space in the aggregate containing the deduplication volume. Use the `volume efficiency on` command to turn on change logging on the volume, which attempts to recreate the deduplication metafiles in the aggregate.

Understanding Compression- and Deduplication-Related Event Messages

This section describes some special events that are logged and related to compression and deduplication.

Table 16) Data compression- and deduplication-related sis log messages.

Event Message	Explanation
sis_undo_stopped	volume efficiency undo operation was stopped by a user.
sis_change log_noload	Previous change logging metafiles could not be reused during initialization. New metafiles will be created.
sis_undo_done	volume efficiency undo –undo-type operation on a volume is completed.
wafl_compress_scan_checkpoint_deleted	Compression of existing data deleted a checkpoint.
sis_op_stopped	Sis operation is stopped by a user.
sis_max_saving_reached	The maximum postprocess compression and deduplication savings on the volume have been reached. No further postprocess compression or deduplication operations will run. If enabled, inline compression will continue to provide additional savings.
sis_max_savings_near	The maximum postprocess compression and deduplication savings on the volume are near. Once you reach this limit no further postprocess compression or deduplication operations will run.
sis_logical_limit_reached	The postprocess compression and deduplication logical data limit on the volume has been reached. No further postprocess compression or deduplication operations will run. If enabled, inline compression will continue to provide additional savings. Logical data is the summation of used blocks, saved blocks, and compressed blocks in the volume.
sis_logical_limit_near	The postprocess compression and deduplication logical data limit on the volume has nearly been reached. Once it is reached no further postprocess compression or deduplication operations will run. If enabled, inline compression will continue to provide additional savings. Logical data is the summation of used blocks, saved blocks, and compressed blocks in the volume.
sis_slow_Notify	The deduplication metadata processed is too big.
sis_change log_drop	The system is unable to change log due to heavy load on the system.
sis_checkpoint_delete	A sis checkpoint was deleted and the current sis operation is restarted from the beginning.
sis_fingerprint_notify_lost	A bad block is encountered while reading deduplication metafiles. As a result, some of the fingerprint records might have been lost. The only impact is potentially lower deduplication savings.
sis_verify_suspended	The system suspended a sis volume-checking operation to run a queued sis operation.
sis_clog_truncate_onrevert	The deduplication change log file is truncated to the maximum size supported by a previous Data ONTAP release during a revert operation.

Location of Logs

The compression and deduplication logs include detailed information about how many data blocks and fingerprints are processed and can be found in `/mroot/etc/log/sis`.

Example: Fri Aug 15 00:43:49 PDT /vol/max1 [sid: 1218783600] Stats (blks gathered 0,finger prints sorted 603349248,dups found 0,new dups found 4882810,blks deduped 0,finger prints checked 0,finger prints deleted 0)

This example reveals the following information:
Total number of new blocks created since the last deduplication process ran = 0
Total number of fingerprint entries (new + preexisting) that were sorted for this deduplication process = 603349248
Total number of duplicate blocks found = 0
Total number of new duplicate blocks found = 4882810
Total number of duplicate blocks that were deduplicated = 0
Total number of fingerprints checked for stale condition = 0
Total number of stale fingerprints deleted = 0

15.8 Additional Compression and Deduplication Reporting

Basic status information for compression and deduplication can be collected using the `volume efficiency show` command with no options. This command will show which volumes have deduplication enabled, the current status, and the progress of compression/deduplication operations.

The following is an example of the output from the basic `volume efficiency show` command.

Cluster1::> `volume efficiency show`

Vserver	Volume	State	Status	Progress
vsone	/vol/volArchive	Enabled	Idle	Idle for 49:59:46
vsone	/vol/volHomeC	Enabled	Idle	Idle for 29:13:39

Note: The `volume efficiency show` command does not currently display information about status and progress for an Infinite Volume. To see the status or progress of individual data constituents, substitute the –volume option for the –is-constituent true option. Ex. `volume efficiency show –vserver <SVMname> -is-constituent true -fields progress`.

Reporting More Details on the Most Recent Compression and Deduplication Operation

To gather more detailed information on the latest compression and deduplication that have been run, use the `volume efficiency show` for a particular volume. This level of detail is only displayed for FlexVol volumes or data constituents, not for the Infinite Volumes as a whole.

To view the same detailed information for Infinite Volume data constituents, you should use the volume efficiency command, substituting the –volume option with the options –is-constituent true and –instance.

Ex. `volume efficiency show –vserver <SVMname> –is-constituent true –instance`

Some of the most common questions that can be addressed with the `volume efficiency show` command include:

- How long the last compression/deduplication operation ran
- How much data was fingerprinted in the change log during the last deduplication operation
- The compression/deduplication schedule

The following is an example of the output from the `volume efficiency show` command run for a particular FlexVol volume, followed by definitions for each value. This information corresponds to the last deduplication operation.

```
Cluster1::> volume efficiency show -vserver vsone -volume volArchive
Vserver Name:                    vsone
Volume Name:                     volArchive
Volume Path:                     /vol/volArchive
State:                           Enabled
Status:                          Idle
Progress:                        Idle for 00:00:26
Type:                            Regular
Schedule:                        sun-sat@0
Efficiency Policy Name:          -
Minimum Blocks Shared:           1
Blocks Skipped Sharing:          0
Last Operation State:            Success
Last Successful Operation Begin: Thu Jul 28 23:45:09 GMT 2011
Last Successful Operation End:   Thu Jul 28 23:45:09 GMT 2011
Last Operation Begin:            Thu Jul 28 23:45:09 GMT 2011
Last Operation End:              Thu Jul 28 23:45:09 GMT 2011
Last Operation Size:             164 GB
Last Operation Error:            -
Change Log Usage:                0%
Logical Data Size:               167 GB
Logical Data Limit:              50.00TB
Logical Data Percent:            0%
Queued Jobs:                     -
Stale Fingerprints:              0%
Compression:                     true
Inline Compression:              true
Incompressible Data Detection:   true
Constituent Volume:              false
Compression Quick Check File Size: 524288000
```

Vserver Name: Name of the Vserver where the volume exists.

Volume Name: Name of the volume.

Volume Path: Absolute path of the volume.

State: Shows if deduplication is enabled or disabled for the volume.

Status: Shows the current state of postprocess compression/deduplication processes: Active, Idle, Initializing, Undoing.

Progress: If the volume's postprocess compression/deduplication operation is in the idle, disabled, or pending state, this shows how long it has been in that state. If the volume's postprocess compression/deduplication operation is active, then it shows the progress of the current operation. As an example, if the operation is in the deduplication sorting phase, this value shows the number of records searched for sorting the fingerprints so far.

Type: Shows the type of the compressed/deduplicated volume, Regular or SnapVault.

Schedule: Shows the postprocess compression/deduplication schedule for the volume.

Efficiency Policy Name: Shows the postprocess compression/deduplication policy for the volume.

Minimum Blocks Shared: The minimum duplicate blocks needed to start sharing. By default it is set to 1.

Blocks Skipped Sharing: The number of blocks that failed to share because the number failed to meet the minimum blocks shared criteria.

Last Operation State: The completion status of the last postprocess compression/deduplication operation.

Last Successful Operation Begin: The time when the last successful postprocess compression/deduplication process (operation) began.

Last Successful Operation End: The time when the last successful postprocess compression/deduplication process (operation) ended.

Last Operation Begin: The time when the last deduplication process (operation) began.

Last Operation End: The time when the last deduplication process (operation) ended.

Last Operation Size: The amount of new data that was processed during the last deduplication operation.

Last Operation Error: The error that occurred, if any, during the last deduplication process (operation).

Change Log Usage: The percentage of how full the change log is. If the change log becomes full, new fingerprints stop being created until the postprocess compression/deduplication operations are run.

Logical Data Size: This value represents the size of logical data in the volume.

Logical Data Limit: This value represents the maximum logical data supported on the system.

Logical Data Percent: This value represents the percentage of logical data in the volume.

Queued Jobs: This number represents the number of postprocess compression/deduplication jobs that are in queue waiting to start. The maximum number of concurrent postprocess compression/deduplication jobs allowed is 8. If there are more than 8 jobs, the subsequent jobs are placed in queue.

Stale Fingerprint Percentage: The percentage of stale records in the fingerprint database.

Compression: Shows if postprocess compression is enabled or disabled for the volume.

Inline Compression: Shows if inline compression is enabled or disabled for the volume.

Incompressible Data Detection: Current state of incompressible data detection (idd) on the volume {enabled or disabled}.

Constituent Volume: Shows if this is a constituent volume that is a part of an Infinite Volume.

Compression Quick Check File Size: Maximum file size (in bytes) to attempt Quick Check on a file. The default value is 500MB.

15.9 Where to Get More Help

For additional support, contact one of the following.

- Your local account team
- Systems engineer
- Account manager
- [NetApp Customer Success Services](#)
- [NetApp Support](#) (formerly NOW®) site
- (888) 4-NETAPP (United States and Canada)
- 00 800 44 NETAPP (EMEA/Europe)
- +800 800 80 800 (Asia/Pacific)

Useful Information to Gather Before Contacting NetApp Customer Success Services

The following compression and deduplication commands and logs provide useful information for troubleshooting the root cause of compression and deduplication issues. This information is very useful when working with NetApp Customer Success Services.

- volume efficiency show -vserver vsone -volume volArchive
- set advanced; volume efficiency check –vserver <SVMname> –volume <volname>
- snap show –vserver <SVMname> -volume <volname>
- node scope snap delta <volname>
- df -h <volname>
- df -S
- All sis logs located in /mroot/etc/log/ - sis, sis.0, sis.1, sis.2, sis.3, sis.4, sis.5
- /mroot/etc/log/messages
- Copy of the EMS logs for the time when the issue is seen

Additional References

- TR-4037: Introduction to NetApp Infinite Volume
  http://www.netapp.com/us/media/tr-4037.pdf
- TR-4070: NetApp Flash Pool Design and Implementation Guide
  http://www.netapp.com/us/media/tr-4070.pdf
- TR-3978: In-Place Expansion of 32-Bit Aggregates to 64-Bit Overview and Best Practices
  http://www.netapp.com/us/media/tr-3978.pdf
- WP-7022: The NetApp Storage Efficiency Guide
  http://www.netapp.com/us/media/wp-7022-1110.pdf
- TR-3742: Using FlexClone to Clone Files and LUNs
  http://www.netapp.com/us/media/tr-3742.pdf
- TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide
  http://www.netapp.com/us/media/tr-3965.pdf

- TR-3450: High-Availability Pair Controller Configuration Overview and Best Practices
  http://www.netapp.com/us/media/tr-3450.pdf
- TR-3747: Best Practices for File System Alignment in Virtual Environments
  http://www.netapp.com/us/media/tr-3747.pdf
- TR-3705: NetApp and VMware VDI Best Practices
  http://www.netapp.com/us/media/tr-3705.pdf
- TR-3770: 2,000-Seat VMware View on NetApp Deployment Guide Using NFS: Cisco Nexus Infrastructure
  http://www.netapp.com/us/media/tr-3770.pdf
- TR-3428: NetApp and VMware Virtual Infrastructure 3 Storage Best Practices
  http://www.netapp.com/us/media/tr-3428.pdf
- TR-3749: NetApp and VMware vSphere Storage Best Practices
  http://www.netapp.com/us/media/tr-3749.pdf
- TR-3886: Understanding and Using vStorage APIs for Array Integration with NetApp Storage
  http://www.netapp.com/us/media/tr-3886.pdf
- TR-3702: NetApp Storage Best Practices for Microsoft Virtualization
  http://www.netapp.com/us/media/tr-3702.pdf
- WP-7053: The 50% Virtualization Guarantee* Program Technical Guide
  http://www.netapp.com/us/media/wp-7053.pdf
- TR-3732: Citrix XenServer and NetApp Storage Best Practices
  http://www.netapp.com/us/media/tr-3732.pdf
- TR-3584: Microsoft Exchange 2007 Disaster Recovery Model Using NetApp Solutions
  http://www.netapp.com/us/media/tr-3584.pdf
- TR-3651: Microsoft Exchange 2007 SP1 Continuous Replication Best Practices Guide
  http://www.netapp.com/us/media/tr-3651.pdf
- TR-3824: Storage Efficiency and Best Practices for Microsoft Exchange Server 2010
  http://www.netapp.com/us/media/tr-3824.pdf
- TR-3979: Oracle on NetApp clustered Data ONTAP Solution Guide
  http://www.netapp.com/us/media/tr-3979.pdf
- TR-3843: Storage Savings with Domino and NetApp Deduplication
  http://www.netapp.com/us/media/tr-3843.pdf.
- NetApp Space Savings Estimation Tool

  https://communities.netapp.com/docs/DOC-18699

Version History

Version	Date	Document Version History
Version 1.0	March 2012	Initial release for compression and deduplication with clustered Data ONTAP 8.1.
Version 1.0.1	June 2012	Updated to include Data ONTAP 8.1.1 updates
Version 2.0	April 2013	Updated to include Data ONTAP 8.2 updates
Version 2.1	February 2014	Updated Flash Cache, Flash Pool and upgrade information

NetApp

www.netapp.com

tApp Data